

IEEE Standard for Media Management System (MMS) Drive Management Protocol (DMP)

Sponsor

**Storage Systems Standards Committee
of the
IEEE Computer Society**

Approved 21 June 2000

IEEE-SA Standards Board

Abstract: Version 1.0 of the Drive Management Protocol (DMP) is specified.

Keywords: application independent, architecture, automated tape library, content neutral, data model, device manager, distributed, drive manager, fully scalable, heterogeneous environment, information access, interoperability, interoperable, language neutral, library management, library manager, media management, media neutral, middleware, opensource, operating system independent, platform-independent, protocol based, removable media, robotic tape library, secure, software, storage, storage management, storage system, system architecture

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 21 August 2000. Printed in the United States of America.

Print: ISBN 0-7381-2512-1 SH94865
PDF: ISBN 0-7381-2513-X SS94865

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

IEEE is the sole entity that may authorize the use of certification marks, trademarks, or other designations to indicate compliance with the materials set forth herein.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

[This introduction is not part of the IEEE Std 1244.4-2000 Standard for Media Management System (MMS) Drive Management Protocol (DMP).] This is one of five MMS standards balloted together under IEEE Project 1244 for Storage System Design.

IEEE Std 1244.1-2000—IEEE Standard for Media Management System (MMS) Architecture

IEEE P1244.2/D041900—Draft Standard for Media Management System (MMS) Session Security, Authentication, Initialization Protocol (SSAIP)

IEEE Std 1244.3-2000—IEEE Standard for Media Management System (MMS) Media Management Protocol (MMP)

IEEE Std 1244.4-2000—IEEE Standard for Media Management System (MMS) Drive Management Protocol (DMP)

IEEE Std 1244.5-2000—IEEE Standard for Media Management System (MMS) Library Management Protocol (LMP)

This MMS DMP standard (IEEE Std 1244.4-2000) is best understood in the context of the others by reading the first two sections of the Architecture (IEEE Std 1244.1-2000). These five standards are the first in a suite of ten approved projects for the IEEE MMS, and all 10 might well have been a single standard. However, their separation was maintained to make conformance easier and individual evolution possible.

By permission of SGI™ this standard is based on the SGI™ OpenVault™ product, which SGI™ has released to the open source community.^a However, the IEEE Media Management System, developed by the IEEE Storage System Standards Working Group (SSSWG), is a significant refinement and extension of OpenVault™. The IEEE Mass Storage System Reference Model (MSSRM) influenced OpenVault™ design before work began on the MMS, and the present OpenVault™ reflects strong influence in its development by the IEEE SSSWG. At the time this standard was completed, the Storage System Standards Working Group (SSSWG) had the following members:

John L. (Jack) Cole, *Chair*

Curtis Anderson
Samuel S. Coleman*
R. Troy Eberlein
Bruce K. Haddon
T. Dixon Hutchinson

Merritt E. Jones*
Jan Klier
Stuart Kreitman
Paul Lockwood

John Merrill
Geoffrey G. Peck
Alan Rollow
Murali Sathyanarana
Joel N. Williams*

* Denotes a charter member of the SSSWG

Technical editors of these five standards are as follows:

IEEE Std 1244.1-2000—IEEE Standard for Media Management System (MMS) Architecture

—Geoffrey G. Peck (Primary Editor for Architecture)

—Curtis Anderson

—Murali Sathyanarayana (Primary Editor for the Data Model)

—Curtis Anderson, Joel N. Williams, T. Dixon Hutchinson, Alan Rollow

^aMore information on OpenVault™ is available at <http://www.ssswg.org>.

IEEE P1244.2/D041900—Draft Standard for Media Management System (MMS) Session Security, Authentication, Initialization Protocol (SSAIP)

—Jan Klier (Primary Editor)
—Bruce K. Haddon, Curtis Anderson, Joel N. Williams

IEEE STD 1244.3-2000—IEEE Standard for Media Management System (MMS) Media Management Protocol (MMP)

—Murali Sathyanarayana (Primary Editor)
—Curtis Anderson, Joel N. Williams, T. Dixon Hutchinson, Alan Rollow

IEEE Std 1244.4-2000—IEEE Standard for Media Management System (MMS) Drive Management Protocol (DMP)

—Joel N. Williams (Primary Editor)
—Murali Sathyanarayana

IEEE Std 1244.5-2000—IEEE Standard for Media Management System (MMS) Library Management Protocol (LMP)

—Joel N. Williams (Primary Editor)
—Murali Sathyanarayana

Sponsor chairs of this standard are as follows:

—Merritt E. Jones (1993-2000)
—John L. (Jack) Cole (2000)

The following members of the balloting committee voted on this standard:

Curtis Anderson	Michael V. Hardy	Ben Kobler
Brian A. Berg	P. C. Hariharan	Stuart K. Kreitman
Paul Buerger	Ronald W. Huff	Paul Lockwood
George B. Cole	T. Dixon Hutchinson	Geoffrey Peck
John L. Cole	Merritt E. Jones	Alan R. Rollow
Don Doerner	Michael P. Kearney	Murali Sathyanarayana
Richard T. Eberlein	Linda Kempster	Bob Snead
Robert J. Gagliano	Charles M. Kennedy	Robert Snively
Mark R. Gary		Joel N. Williams

IEEE Standard for Media Management System (MMS) Drive Management Protocol (DMP)

1. Overview

1.1 Scope

This standard describes the syntax and semantics of the protocol messages that pass between the Drive Manager (DM) and the Media Manager (MM). The Library Manager (LM) exists only in the context of a larger Media Management System (MMS); a thorough understanding of the architecture of the MMS as described in IEEE Std 1244.1-2000 is a prerequisite. In addition, certain elements common to all of the modules in the MMS are described in P1244.2/D041900.¹ The MM itself, a system that operates as a peer process with the LM, is described in IEEE Std 1244.3-2000.

1.2 Purpose

This standard specifies the protocol and meaning of messages that pass between the DM and the MM of an IEEE MMS. The language that the MM module uses to communicate with the DM, and the corresponding language that the DM uses to communicate with the MM, are collectively called the Drive Management Protocol (DMP). This protocol determines and/or describes all relevant behavior and states of the MM and the DM. The standardized protocol and semantics of the DMP allow any conforming MM and DM to operate together as components of a single MMS.

1.3 Conformance

A DM conforms to this standard if it can send and receive protocol messages in the syntax described in this standard, acts in accordance with the semantics of those messages, and operates the underlying device(s) as described in this standard and the related standards mentioned in 1.1. Those protocol messages described in this standard shall be implemented exactly. No additional protocol messages, that is, messages not defined in this standard, may be added. Exactly those objects, as they relate to the DM, described in the database in IEEE Std 1244.1-2000 shall be supported, and no additional objects may be added.

¹Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining drafts, contact the IEEE.

This standard specifies version 1.0 of the DMP. It is expected that this protocol will periodically be revised, at which time addenda to or revisions of this standard will be issued. DMs and MMs negotiate for a mutually agreeable version of the DMP as described in P1244.2/D041900.

1.4 Rationale

The DMP provides a common drive interface for the MMS, abstracting all of the details of drive configuration and control. Thus a common drive interface is established for all drives in the MMS. This allows for the *plug and play* of new managed drives, i.e., the addition of a managed drive without recompilation of any of the modules in the Media Manager System.

2. References, definitions

A suite of IEEE 1244 standards describes the MMS. This modular approach to standardizing a MMS permits a granularity of conformance not possible otherwise, and allows a degree of independent evolution for the components of an IEEE MMS. Nevertheless, the overall architecture of the IEEE MMS is described in IEEE Std 1244.1-2000 and fundamental information (including the data model) that is common to all related standards is communicated through that standard. Reference to the MMS Architecture is crucial in understanding the DMP.

This standard shall be used in conjunction with the following standards. When the following standards are superseded by an approved revision, the revision shall apply.

IEEE Std 1244.1-2000, IEEE Standard of Media Management System (MMS) Architecture.²

P1244.2/D041900, Draft Standard for Media Management System (MMS) Session Security, Authentication, Initialization Protocol (SSAIP).³

IEEE Std 1244.3-2000, IEEE Standard for Media Management System (MMS) Media Management Protocol (MMP).

IEEE Std 1244.5-2000, IEEE Standard for Media Management System (MMS) Library Management Protocol (LMP).

Additional references common to all IEEE MMS standards can be found in Clause 2 of IEEE Std 1244.1-2000; definitions can be found in Clause 3 of IEEE Std 1244.1-2000.

3. DM definition and characteristics

3.1 Basic functions of the DM

The DM module is the MMS component that manages the configuration of drives and handles drive control tasks associated with MM-mount and MM-unmount requests for drives that are under MMS management.

²IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://standards.ieee.org/>).

³This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.

There is at least one DM associated with each MMS-managed drive. The main purpose of the DM is to expose the drive configuration to the MM and optionally to control the drive. A DM may only support or be used by one MM.

If there is a DM that implements drive control in addition to exposing drive configuration, then the drive is said to have an MMS-accessible control interface. In this case, the MMS can access drive control through that DM. MMS-managed drives may include those both with and without an MMS-accessible control interface.

3.2 System configuration for a DM

The DM shall maintain the configuration attributes for its drive that are listed in Table 1. These attributes shall be maintained persistently, such as in a configuration file on the machine where the DM resides.

Table 1—Configuration attributes for a DM

MM communications address	The name of the host (or IP address) and port numbers where the MM can be found. This information is in the DM configuration in order to enable the DM upon boot to initiate contact with the MM.
Drive name	The name by which the MMS knows the drive. Drive names are unique within the context of a single MMS. The MMS uses this information to detect multi-hosted drives. For multi-hosted drives, each controlling DM has the same drive name.
Physical type	Describes physical properties like cartridge form factor (DLT, 4 mm, 8mm, etc.) and media interchangeability (DLT2000, DLT4000, 8 mm–8500, etc.). There is a canonical list of these properties.
Access method mapping	The configuration file may define the mapping between each legal mode of access and the drive handle and/or the local access method used to accomplish that mode of access.

3.3 Drive control constraints

Although only one DM may be managing a drive at any time, it is possible to have multiple DMs, with one active and others in reserve in case of a failure of the active manager. The MM ensures that only one DM at a time is actively controlling a library. If the MM switches the controlling DM, it sends an *activate disable* command to the old controlling DM to force it to stop communicating with the drive hardware; then it sends an *activate enable* command to the new controlling DM to force it to resynchronize with the drive hardware.

A DM is required to start itself immediately whenever the machine on which it runs is started. Hence if the machine crashes, or the DM goes down for some other reason, then it must restart itself automatically. The restarted DM may then be activated by the MM. In the case of a failure of one DM, the MM shall not activate another DM until the one that has failed has had sufficient time to restart itself.

If a DM is restarted after a machine crash, or for some other reason, the MM will ask the DM to detach any drive handles that are no longer in use.

See the **detach**, **activate**, and **ready** commands for details.

3.4 Implementation constraints and considerations

Most commonly, drives are host-attached and a DM on a host to which the drive is attached implements drive control through a host interface to the drive. A drive that is attached to multiple hosts may have multiple MMS-accessible control interfaces and associated DMs: one for each host to which that drive is attached.

Typically, if a drive is not host-attached, so that a DM cannot control it through a host interface, the DM does not implement drive control, so there would be no MMS-accessible control interface to such a drive. An example of a drive that may lack an MMS-accessible control interface is a video drive that is similar to a VCR machine, where the only way to control the drive is through an external control panel with stop, play, rewind, forward, and pause control buttons.

Design goals for the DM are to allow implementations for any of the following:

- a) Physical type of removable media drive
- b) Device interface protocol (i.e., channel-, bus-, and network-attach protocols)
- c) Operating system

Since the drive interface consists of text messages over TCP/IP, which is supported by all systems, this goal is feasible.

3.5 State and fault tolerance

A key MMS goal that impacts the strategy of its slave control programs for managing the state is that a control program should maintain a minimal persistent state in order to accelerate crash recovery for control programs. A DM-specific goal for fault tolerance is that the DM should avoid interfering with application access to the drive whenever possible, including across DM reboots. The DM strategy for managing internal state attempts to balance these two somewhat conflicting goals. In order to support continuous application access to the drive, the DM must maintain a persistent state.

Consider that once a DM has a drive handle for a host-attached drive, an MMS Client Application may establish access to the drive independent of the state of the DM. Even if the DM crashes, so long as the application/DM-host remains up and the drive control and data paths remain available, the application may continue to access the drive. The DM should not interrupt this access unnecessarily.

There is a back-channel protocol from the DM to the MMS, called DMP/D, which includes the means to create/access persistent attributes stored in the MMS database. When a DM boots, it queries the MM database to recover any state it deems necessary from those attributes.

3.6 Abstraction of a drive

3.6.1 Capabilities and mode of access

Each drive has an associated set of capabilities, each of which describes a specific settable feature that is supported by the DMP abstraction of the drive. (See IEEE Std 1244.1-2000 for a description of how these capabilities are represented in the database). These capabilities determine such things as which device driver to use for control and data access, what device settings to select, and what device state to check. A particular combination of capabilities represents a particular mode of access. Each drive has a configurable set of legal modes of access, which may depend on the media types that may be accepted by the drive. The configurable set of modes includes a default mode of access.

Each legal mode of access represents a logical instance of the drive with underlying control and access methods. The default mode of access for a drive depends at least in part on the type of drive (e.g., audio vs data, etc.), and possibly the type of media in the drive (e.g., EXA8200 tape in an EXA8500 drive). Default modes are defined as well-known tokens, so that they are well understood by all applications.

The use of these capabilities and modes of access is what permits DM to hide implementation details such as the underlying local control and access methods for the device.

3.6.2 Drive components

3.6.2.1 Mount point

This is the physical drive opening where a cartridge may be placed, often called the drive door. A cartridge must be present at the mount point before the cartridge can be moved to and the media it contains engaged at the media access point. When the media is disengaged from the media access point (and returned to its cartridge, as necessary), the cartridge is returned to the mount point.

3.6.2.2 Media access point

This is the component that reads and writes the media contained in a cartridge. A cartridge must be moved to and the media it contains engaged at a media access point before the media can be accessed for a read or write operation. The cartridge, its media, or the media access point may physically restrict access to the media. For instance, physical access to the media may be restricted to read-only.

Once media is engaged at the media access point, media data may be accessed via the drive data path, if one exists.

3.6.2.3 Control path

This is a control access interface between the DM host and the drive. It is accessible by the DM, and possibly by MMS Client Applications on the DM host. Typically, underlying this interface is a physical connection, such as a local channel, bus, or shared network between the DM host and the drive.

3.6.2.4 Data path

This is a data access interface between the DM host and the drive media access point. It is accessible by the DM, and by MMS Client Applications on the DM host. Drives that have a control path typically also have a data path, with the control and data paths sharing the same physical connection, and with access through a local device driver. For drives with a data path, the DM may require access to the drive data path in order to control the drive, (e.g., to identify a partition).

A drive media access point may lack a data path. An example of a media access point that lacks a host connection is a set of RGB lines that attach a video drive to a display device. Applications will not have access to this type of media access point.

3.6.2.5 MMS-accessible control interface

Typically, an MMS-accessible control interface is composed of some combination of the drive control and data paths, depending on the drive and its host connectivity. If the DM does not have access to one or both of the control and data paths, certain DMP commands may be implemented as no-ops. An implementer of a DM may control a drive by other means than control and data paths, (e.g., by interacting with an operator or a separate software control system).

3.6.2.6 Media

This is a recordable surface upon which data resides. A cartridge may contain one or more pieces of media. Associated with a piece of media and a drive is a media type, which determines the size of the recording surface; and a bit format, which determines the recording format.

3.6.2.7 Partition

Some drives support the partitioning of media. A partition corresponds to a specific region on the media that has a physically marked beginning and ending, and is only accessible from the media access point (e.g., a tape partition). The MMS will allow applications to mount a particular soft partition. DMP supports this by allowing the MMS to request that the media contained by a cartridge be positioned to a particular soft partition. For drives that do not support partitioning of media, the entire recordable media surface is viewed as a single soft partition.

Note that while the DMP provides the required interface, not all device drivers support soft partitions.

3.6.3 Local instantiation

For drives that have an MMS-accessible control interface, the drive must be instantiated on the DM host before the DM can access the drive. Typically, this involves creating a local binding between a name (i.e., device file pathname) and a logical instance of the drive (i.e., dev node) that corresponds to a particular mode of access. The name is called a drive handle. For drives that have a data path, the drive handle may be passed to and used by an MMS Client Application to send drive control or access media at the media access point. When the binding is removed, the drive handle is invalidated.

4. DMPs overview

The communication between the MM and the DM uses the full-duplex, bi-directional, peer-to-peer interface described in IEEE Std 1244.1-2000. As noted in that standard, the protocol consists of three phases: command, acknowledgement, and response. This standard does not describe that interface, but rather describes the syntax and semantics of the messages that are transmitted across that interface.

There are two distinct languages used for communication between the DM and the MM, one for each direction in the bi-directional connection. They are as follows:

- a) DMP/M: the Library Management Protocol language spoken by the MM to the DM.
- b) DMP/D: the Library Management Protocol language spoken by the DM to the MM.

One module *speaks* to another if it initiates the command phase of the protocol. In this standard, DMP refers to the DMP, which is made up of both DMP/M and DMP/D.

The DMP/D and DMP/M implement different command sets, reflecting their different requirements. The DMP/M language defines a drive control interface, while the DMP/D language defines primarily a status reporting, administration, and configuration interface.

The relationship between the MM and a DM is a peer-to-peer relationship. The DM has the ability to store and retrieve persistent data in the MMS database via the DMP/D language, and to inform the MMS of changes in the state of the drive.

The DMP utilizes the task identifier mechanism common to all MMS protocols, as described in IEEE Std 1244.1-2000. This mechanism directly supports the concurrent execution of multiple tasks. For a drive, there

is little, if any, need for concurrent execution of multiple tasks, and there is no requirement to support that feature; but the mechanism is nevertheless present.

An administrative application, specifically written for a particular DM, and running as a client of the MM, may use the setting and reading of special named attributes as a private means of interaction or control. In this case, commands would be passed from the administrative application through the MM to the DM. See IEEE Std 1244.1-2000 for directions on how to create such special attributes. Such named attributes in a DM may be thought of as *control registers* that cause an operation to occur as a side effect of setting or reading those attributes.

The command definitions that follow in Clause 7 and Clause 8 are in Extended Backus-Naur Format (EBNF) as described in IEEE Std 1244.1-2000. Note that all undefined non-terminals resolve to a string, but that for purposes of brevity, this detail is left out. In addition, when these resolve to an entry in the database, this is noted in the subclauses denoted *identifier arguments*.

5. Response and error message format for DMP

Components of the protocol sequence, as described in IEEE Std 1244.1-2000, are shown in Table 2.

Table 2—Protocol sequence

Command	Described in Clause 7 and Clause 8.
Acknowledgement	The content and format of the acknowledgement is described in IEEE Std 1244.1-2000. It indicates that the receiving program has or has not successfully received and parsed the command.
Response	The format of the response is described in IEEE Std 1244.1-2000. As noted in that standard, if there are errors in the execution of the command, these errors are returned with the response. In addition, some commands, when executed successfully, contain additional parameters in the response. The specific error codes for this protocol for each command are described with the command in Clause 7 and Clause 8. Error codes that apply to all commands are described in Clause 9. The additional parameters in the response, if any, are listed for each command in Clause 7 and Clause 8.

As noted in IEEE Std 1244.1-2000, the response may include an error message in the format of a message that may be represented in a local language. The EBNF for that message is:

```
<message-clause>:: = "message" "[" { <id-clause> "[" <arg-clause> "]"
                                <locale-text-clause>* } "]"
<id-clause> ::= "id" "[" <manufacturer> <model> <messageid> "]"
<arg-clause> ::= "arguments" "[" <arg-pair>+ "]"
<arg-pair> ::= <arg-key> <arg-text>
<locale-text-clause> ::= "loctext" "[" <language> <localized-string> "]"
```

Messages of this type are used in contexts other than the response, as noted in clauses to follow.

The EBNF for the **response** is:

```
<response-message> ::= "response" { <negative-response> |  
                                     <accepted-response> |  
                                     <command-response> } ";"  
  
<negative-response> ::= { "unacceptable" [ <message-clause> ] }  
<accepted-response> ::= { <task-clause> "accepted"  
<command-response> ::= { <task-identifier  
                           <final-response> [ <message-clause> ] }  
  
<final-response> ::= <response-success> |  
                     <response-cancelled> |  
                     <response-error>  
  
<response-success> ::= { "success" <response-text>* }  
<response-cancelled> ::= "cancelled"  
<response-text> ::= text "[" <text-string>+ "]"  
<task-clause> ::= task "[" <task-identifier> "]"  
<response-error> ::= error "[" <class> <code> "]"
```

6. DMP/M command descriptions

6.1 Command name: attach

The **attach** command is supported for all MMS-managed drives, but is a no-op for drives that lack an MMS-accessible control interface, and may be a no-op if no data path underlies this interface.

The attach command selects the appropriate logical instance of the drive according to the mode of access that is specified by the MMS. It configures a logical instance of the drive, as necessary, and binds an opaque drive handle to this logical instance (i.e., in a UNIX[®] environment, this equates to creating or linking to a dev node). The DM generates the drive handle; it must be unique on the DM host.

In the case where the drive supports partitioned media (i.e., a CDROM or a high-end tape), the drive handle that will be created may be dependent on the partition. For example, most disks on UNIX[®] platforms have the partition to be accessed encoded in the drive handle (i.e., dev node) that is created. The loaded media will be positioned to the specified partition. For drives that do not support partitioned media, a request to position to other than the first one will return an error.

Attaching to the first partition means positioning to the physical beginning of the recordable media surface (i.e., for tape, this is a rewind to BOT). See Pre-Defined Tokens and Descriptions in IEEE Std 1244.1-2000 for more examples of standardized partition names.

Note that positioning the media may require the DM to have access to a data path to the drive, and otherwise may not be supported.

Typically, a drive has a shared control/data path. In this case, the drive handle that is passed back to the MMS is ultimately passed back to an MMS Client Application. The application uses the drive handle to establish access to the drive control/data path.