



Specification

DSP0124

STATUS: Preliminary

Copyright © "2000" Distributed Management Task Force, Inc. (DMTF). All rights reserved. DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. DMTF specifications and documents may be reproduced for uses consistent with this purpose by members and non-members, if correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release cited should always be noted."

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

June 5, 2001

Abstract

This document presents an LDAP schema for the CIM version 2.5 Physical Information Model [[1](#)].

Change History

Version 1.0	May 21, 2001	Initial Version
-------------	--------------	-----------------

Editor

Ryan Moats
For the DMTF LDAP Mapping Working Group

1 Introduction

This document presents an LDAPv3 [[2](#), [3](#), [5](#)] schema for the DMTF CIM Physical 2.5 Model [[1](#)]. This mapping follows the methodology used for the Core Mapping document [[4](#)] except as noted below in that abstract CIM classes are mapped to abstract LDAP classes. Concrete CIM classes are mapped to structural and auxiliary LDAP classes. CIM associations are mapped using a combination of auxiliary classes and structural LDAP classes.

The content, naming and structure rules provided here are suggestions and may be modified as needed to support a particular directory structure. In addition, the attribute and object class descriptive fields are provided for human clarity. Directory administrators do not need to subclass/instantiate everything in this schema verbatim. They are free to choose the subset that meets their particular needs. In particular, this means:

- If your directory implementation does not support content, naming or structure rules, comment them out.
- If your directory implementation does not support description fields of greater than some length, those may be commented out.
- If your directory implementation requires a certain ordering of classes and attributes (e.g. to avoid forward references) feel free to re-order as necessary.
- If your directory implementation and application can make use of substring matching, feel free to add substring matching clauses as appropriate.

This document is a product of the DMTF LDAP Mapping WG.

2 LDAP Mapping Considerations

2.1 Existing LDAP Classes and Attributes

In addition the following table states the mapping between CIM properties and LDAP attributes along with the reference for the definition of that particular LDAP attributes.

CountryDesignation	cc
--------------------	----

2.2 Mapping Associations

For ease of reference, the following text has been copied from 2.6.2 of the Core Mapping document:

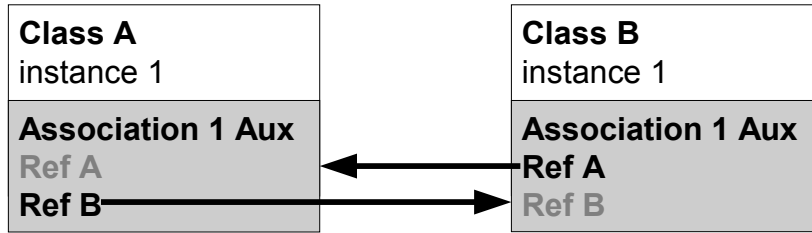
There are three distinct models used for mapping non-abstract associations in this document. Each has its own conventions for how such associations are not only mapped, but also implemented in the directory. The following sections discuss these conventions.

Since all associations have referential properties, the term “additional properties” in the remainder of this section refers to non-referential properties. The approach in Section 2.2.3 may also be used to map associations with no additional properties and 1-to-1 or 1-to-many associations with additional properties if necessary.

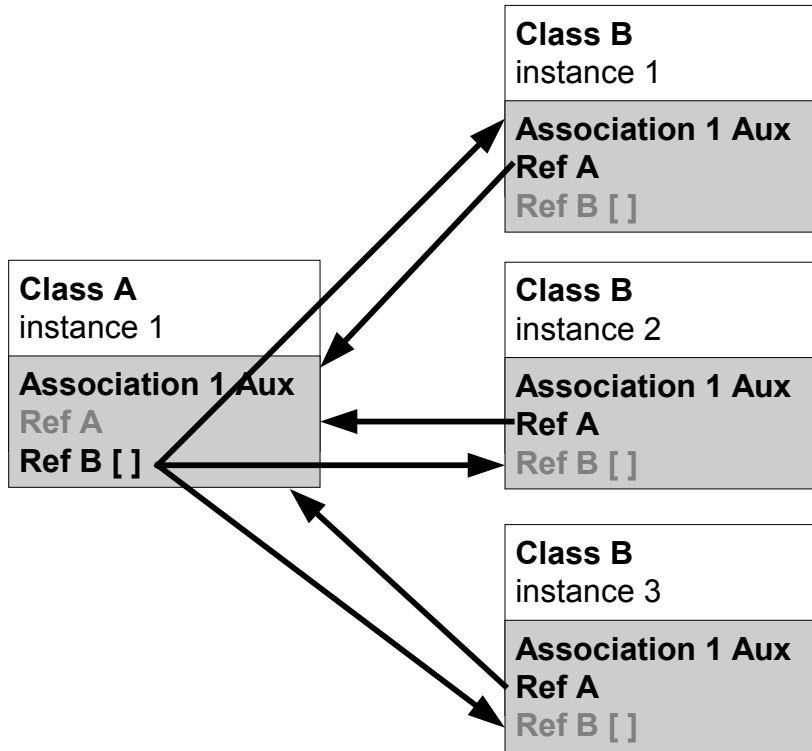
2.2.1 No Additional Properties

If a non-abstract association has no additional properties, then it is mapped as an auxiliary class that contains both referential properties as optional DN attributes. This class is attached to all structural objects that participate in the association, with the proper attribute being populated for

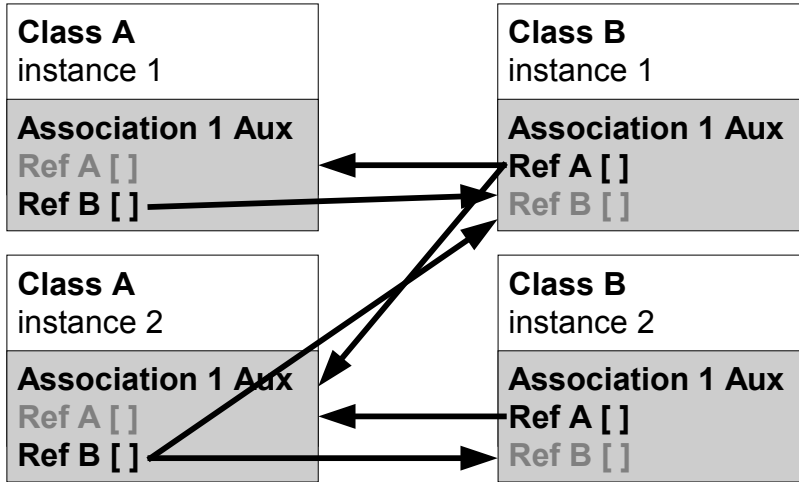
that particular structural object. An example of this type of association is CIM_HostedService.



One-to-One



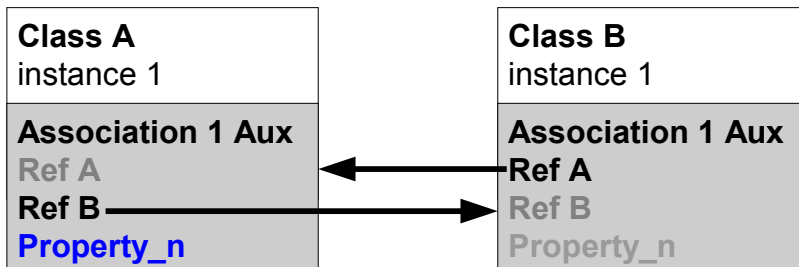
One-to-many



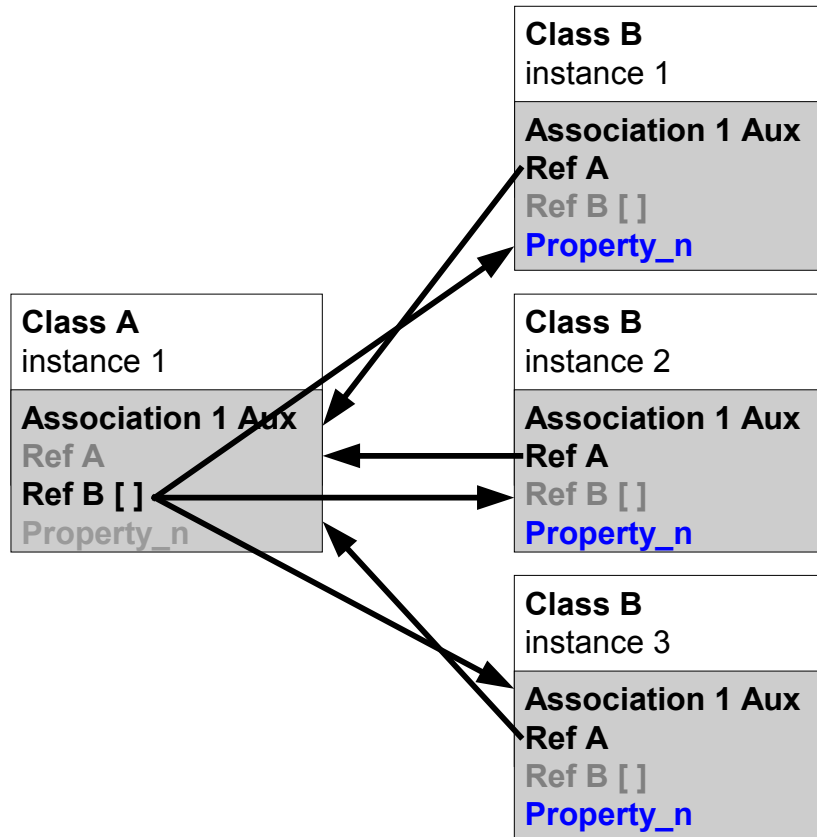
Many-to-Many

2.2.2 Additional Properties, 1-to-1 or 1-to-many

If a non-abstract association has additional properties, then the mapping is determined by the cardinality of the referential properties. In the case of a 1-to-1 or 1-to-many cardinality, the association is mapped as an auxiliary class with all properties mapped as optional attributes. The auxiliary class is attached to all structural objects participating in the association, with the referential attribute set appropriately. The additional properties are set for the auxiliary class that is attached to the many side of a 1-to-many association.



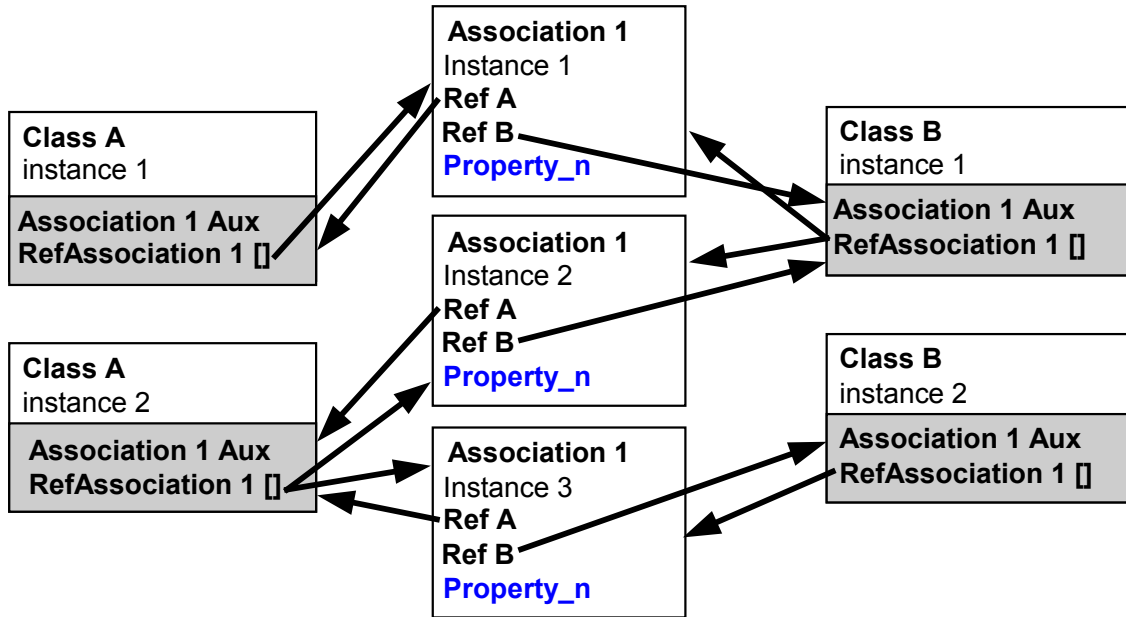
One-to-one with properties



One-to-many with properties

2.2.3 Additional Properties, many-to-many

For a non-abstract association with additional properties and a many-to-many cardinality, the most flexible mapping is to use a structural LDAP class that contains all properties of the association as optional attributes. Since this is a separate object in the directory, helper auxiliary classes are provided that are attached to the structural objects in the directory participating in the association. These helper classes contain a single optional attribute that points to the particular instance of the association that this object participates in. There is an instance of the structural class for every instance of the association. An example of this type of association is `CIM_ServiceServiceDependency`.



Many-to-many with properties

This approach may also be used to map associations with no additional properties and 1-to-1 or 1-to-many associations with additional properties if necessary.

2.3 Helper Classes

When CIM defines linked indexed arrays, the values for these arrays are replaced with separate instances of helper classes that are DIT contained by the original class. Since LDAP does not provide a way to specify DIT containment for entries below an auxiliary class, we point out that any helper class is intended to be attached to an entry, be it structural or auxiliary flavor.

2.3.1 dlmServicePhilosophyInstance

The class dlm1PhysicalFrame defines two linked indexed arrays: ServicePhilosophy and ServiceDescription. These are replaced with separate instances of dlmServicePhilosophyInstance.

```
( 1.3.6.1.4.1.412.100.3.2.1015 NAME 'dlmServicePhilosophy'
DESC 'ServicePhilosophy is an enumerated,
integer-valued array that indicates whether the Frame
is serviced from the top (value=2), front (3), back
(4) or side (5), whether it has sliding trays (6) or
removable sides (7), and/or whether the Frame is
moveable (8), for example, having rollers. Values are
0="Unknown", 1="Other", 2="Service From Top Service
From Front", 3="Service From Back", 4="Service From
Side", 5="Sliding Trays", 6="Removable Sides",
7="Moveable".'
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1014 NAME 'dlmServiceDescriptions'
DESC 'An array of free-form strings providing more
detailed explanations for any of the entries in the
Service Philosophy array. Note, each entry of this
array is related to the entry in ServicePhilosophy
that is located at the same index.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.138 NAME 'dlmServicePhilosophyInstance'
DESC 'helper class to tie ServicePhilosophy and
ServiceDescriptions in PhysicalFrame together'
SUP top
MUST ( arrayIndex )
MAY ( dlmServicePhilosophy $ dlmServiceDescription )
)

( 1.3.6.1.4.1.412.100.3.3.2.25 NAME
'dlmServicePhilosophyInstanceNameForm'
OC dlmServicePhilosophyInstance
MUST ( arrayIndex )
)

( <physical-sr-25> NAME 'dlmServicePhilosophyInstanceStructureRule'
FORM dlmServicePhilosophyInstanceNameForm
SUP ( <physical-sr-2> )
)
```

2.3.2 dlmChassisTypeInstance

The class `dlm1Chassis` defines two linked indexed arrays: `ChassisTypes` and `TypeDescriptions`. In the LDAP mapping, these are replaced with separate instances of `dlmChassisTypeInstance`.

```
( 1.3.6.1.4.1.412.100.3.2.1019 NAME 'dlmChassisTypes'
DESC 'An enumerated, integer-valued array indicating
the type of Chassis. Values are 1="Other",
2="Unknown", 3="Desktop", 4="Low Profile Desktop",
5="Pizza Box", 6="Mini Tower", 7="Tower",
8="Portable", 9="LapTop", 10="Notebook", 11="Hand
Held", 12="Docking Station", 13="All in One", 14="Sub
Notebook", 15="Space-Saving", 16="Lunch Box", 17="Main
System Chassis", 18="Expansion Chassis",
19="SubChassis", 20="Bus Expansion Chassis",
21="Peripheral Chassis", 22="Storage Chassis",
23="Rack Mount Chassis", 24="Sealed-Case PC".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1023 NAME 'dlmTypeDescriptions'
DESC 'An array of free-form strings providing more
```

```

        information on the ChassisTypes array entries. Note,
        each entry of this array is related to the entry in
        ChassisTypes that is located at the same index.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.139 NAME 'dlmChassisTypeInstance'
DESC 'helper class to tie ChassisType and
TypeDescriptions in Chassis together'
SUP top
MUST ( arrayIndex )
MAY ( dlmChassisType $ dlmTypeDescription )
)

( 1.3.6.1.4.1.412.100.3.3.2.26 NAME 'dlmChassisTypeInstanceNameForm'
OC dlmChassisTypeInstance
MUST ( arrayIndex )
)

( <physical-sr-26> NAME 'dlmChassisTypeInstanceStructureRule'
FORM dlmChassisTypeInstanceNameForm
SUP ( <physical-sr-4> )
)

```

2.3.3 dlmMediaTypesSupportedInstance

The class `dlm1StorageMediaLocation` defines two linked indexed arrays: `MediaTypesSupported` and `MediaSizesSupported`. In the LDAP mapping, these are replaced with separate instances of `dlmMediaTypeSupportedInstance`.

```

( 1.3.6.1.4.1.412.100.3.2.1035 NAME 'dlmMediaSizesSupported'
DESC 'The sizes (in inches) of the particular
MediaTypes that may be placed in the Location. Note,
each entry of this array is related to the entry in
the MediaTypesSupported array that is located at the
same index. The value is considered to be Inches.'
SUP cimFloat32
)

( 1.3.6.1.4.1.412.100.3.2.1036 NAME 'dlmMediaTypesSupported'
DESC ' Certain StorageMediaLocations may only be able
to accept a limited set of PhysicalMedia MediaTypes.
This property defines an array containing the types of
Media that are acceptable for placement in the
Location. Additional information and description of
the contained MediaTypes can be provided using the
TypesDescription array. Also, size data (for example,
DVD disc diameter) can be specified using the
MediaSizesSupported array. Values defined here
correspond to those in the Physical Media.MediaType
property. This allows quick comparisons using value
equivalence calculations. It is understood that there
is no external physical difference between (for

```


DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
example) DVDVideo and DVD-RAM. But, equivalent values
in both the Physical Media and StorageMediaLocation
enumerations allows for one for one comparisons with
no additional processing logic (i.e., the following is
not required ... if "DVD-Video" then value="DVD").
Values are 0="Unknown", 1="Other", 2="Tape Cartridge",
3="QIC Cartridge", 4="AIT Cartridge", 5="DTF
Cartridge", 6="DAT Cartridge", 7="8mm Tape Cartridge",
8="19mm Tape Cartridge", 9="DLT Cartridge",
10="Half-Inch Magnetic Tape Cartridge", 11="Cartridge
Disk", 12="JAZ Disk", 13="ZIP Disk", 14="SyQuest
Disk", 15="Winchester Removable Disk", 16="CD-ROM",
17="CD-ROM/XA", 18="CD-I", 19="CD Recordable",
20="WORM", 21="Magneto-Optical", 22="DVD",
23="DVD-RW+", 24="DVD-RAM", 25="DVD-ROM",
26="DVD-Video", 27="Divx", 28="Floppy/Diskette",
29="Hard Disk", 30="Memory Card", 31="Hard Copy",
32="Clik Disk", 33="CD-RW", 34="CD-DA", 35="CD+",
36="DVD Recordable", 37="DVD-RW", 38="DVD-Audio",
39="DVD-5", 40="DVD-9", 41="DVD-10", 42="DVD-18",
43="Magneto-Optical Rewriteable", 44="Magneto-Optical
Write Once", 45="Magneto-Optical Rewriteable
(LIMDOW)", 46="Phase Change Write Once", 47="Phase
Change Rewriteable", 48="Phase Change Dual
Rewriteable", 49="Ablative Write Once", 50="Near Field
Recording", 51="MiniQic", 52="Travan", 53="8mm Metal
Particle", 54="8mm Advanced Metal Evaporate",
55="NCTP", 56="LTO Ultrium", 57="LTO Accelis", 58="9
Track Tape", 59="18 Track Tape", 60="36 Track Tape",
61="Magstar 3590", 62="Magstar MP", 63="D2 Tape",
64="Tape - DST Small", 65="Tape - DST Medium",
66="Tape - DST Large".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)
( 1.3.6.1.4.1.412.100.3.1.2.140 NAME 'dlnMediaTypesSupportedInstance'
DESC 'helper class to tie MediaTypesSupported and
MediaSizesSupported in StorageMediaLocation together'
SUP top
MUST ( arrayIndex )
MAY ( dlnMediaTypesSupported $ dlnMediaSizesSupported $
dlnTypeDescriptions )
)
( 1.3.6.1.4.1.412.100.3.3.2.27 NAME
'dlnMediaTypesSupportedInstanceNameForm'
OC dlnMediaTypesSupportedInstance
MUST ( arrayIndex )
)
( <physical-sr-27> NAME 'dlnMediaTypesSupportedInstanceStructureRule'
FORM dlnMediaTypesSupportedInstanceNameForm
SUP ( <physical-sr-7> )
)
```

2.3.4 dlmPhysicalLabelsInstance

The classes dlm1PhysicalMedia and dlm1Magazine define three linked indexed arrays: PhysicalLabels, LabelStates, and LabelFormats. In the LDAP mapping, these are replaced with separate instances of dlmPhysicalLabelsInstance.

```
( 1.3.6.1.4.1.412.100.3.2.1037 NAME 'dlmLabelFormats'
DESC 'An array of enumerated integers describing the
      formats of each of the labels on a Magazine. The
      Labels themselves are listed in the PhysicalLabels
      property. Note, each entry of this array is related to
      the entry in PhysicalLabels that is located at the same
      index. Values are 0="Barcode", 1="Radio Frequency
      Identification", 2="OCR (Optical Character
      Recognition)", 3="MICR (Magnetic Ink Character
      Recognition)", 4="7 Character Barcode", 5="9 Character
      Barcode".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1038 NAME 'dlmLabelStates'
DESC 'An array of enumerated integers describing the
      states of each of the labels on a Magazine. The Labels
      themselves are listed in the PhysicalLabels property.
      Note, each entry of this array is related to the entry
      in PhysicalLabels that is located at the same index.
      Values are 0="OK/Readable", 1="Unreadable", 2="Upside
      Down".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1039 NAME 'dlmPhysicalLabels'
DESC 'One or more strings on "labels" on the Magazine.
      The format of the labels and their state (readable,
      unreadable, upside-down) are indicated in the
      corresponding LabelFormats and LabelStates array
      properties.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.141 NAME 'dlmPhysicalLabelsInstance'
DESC 'helper class to tie PhysicalLabels, LabelStates, and
      LabelFormats in PhysicalMedia together'
SUP top
MUST ( arrayIndex )
MAY ( dlmPhysicalLabels $ dlmLabelStates $ dlmLabelFormats )
)

( 1.3.6.1.4.1.412.100.3.3.2.28 NAME
'dlmPhysicalLabelsInstanceNameForm'
OC dlmPhysicalLabelsInstance
MUST ( arrayIndex )
)
```

```
( <physical-sr-28> NAME 'dlmPhysicalLabelsInstanceStructureRule'
  FORM dlmPhysicalLabelsInstanceNameForm
  SUP ( <physical-sr-8> $ <physical-sr-12> )
)
```

2.4 Syntax Conversion

The physical model includes attributes that require mapping floating point attributes. Mapping of these attributes is accomplished by inheriting from the attributes `cimFloat32` and `cimFloat64`, defined in [6]. Interested readers are directed there for information about these attribute definitions.

3 Class Definitions

For efficiency in the LDAP representation, associations are specified as a combination of auxiliary classes and DIT structure rules. Attribute definitions for each class are presented with the object class. Other definitions are also provided when necessary.

While this approach was chosen to minimize the number of DN pointers stored in the schema, some pointer dereferencing may be necessary.

3.1 PhysicalPackage Classes

These classes model `PhysicalElements` that contain or host other components. Examples of this are a Rack enclosure or an adapter Card.

```
( 1.3.6.1.4.1.412.100.3.2.1001 NAME 'dlmDepth'
  DESC 'The depth of the PhysicalPackage in inches. The
        value is considered to be Inches.'
  SUP cimFloat32 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1002 NAME 'dlmHeight'
  DESC 'The height of the PhysicalPackage in inches. The
        value is considered to be Inches.'
  SUP cimFloat32 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1003 NAME 'dlmHotSwappable'
  DESC 'A PhysicalPackage is HotSwappable if it is
        possible to replace the Element with a physically
        different but equivalent one while the containing
        Package has power applied to it (ie, is "on"). For
        example, a disk drive Package inserted using SCA
        connectors is both Removable and HotSwappable. All
        HotSwappable packages are inherently Removable and
        Replaceable.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1004 NAME 'dlmRemovable'
  DESC 'A PhysicalPackage is Removable if it is designed
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

to be taken in and out of the physical container in which it is normally found, without impairing the function of the overall packaging. A Package can still be Removable if power must be "off" in order to perform the removal. If power can be "on" and the Package removed, then the Element is both Removable and HotSwappable. For example, an extra battery in a laptop is Removable, as is a disk drive Package inserted using SCA connectors. However, the latter is also HotSwappable. A laptop's display is not Removable, nor is a non-redundant power supply. Removing these components would impact the function of the overall packaging or is impossible due to the tight integration of the Package.'

SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE

)

(1.3.6.1.4.1.412.100.3.2.1005 NAME 'dlmReplaceable'

DESC 'A PhysicalPackage is Replaceable if it is possible to replace (FRU or upgrade) the Element with a physically different one. For example, some ComputerSystems allow the main Processor chip to be upgraded to one of a higher clock rating. In this case, the Processor is said to be Replaceable. Another example is a power supply Package mounted on sliding rails. All Removable packages are inherently Replaceable.'

SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE

)

(1.3.6.1.4.1.412.100.3.2.1006 NAME 'dlmWeight'

DESC 'The weight of the PhysicalPackage in pounds. The value is considered to be Pounds.'

SUP cimFloat32 SINGLE-VALUE

)

(1.3.6.1.4.1.412.100.3.2.1007 NAME 'dlmWidth'

DESC 'The width of the PhysicalPackage in inches. The value is considered to be Inches.'

SUP cimFloat32 SINGLE-VALUE

)

(1.3.6.1.4.1.412.100.3.1.2.1 NAME 'dlm1PhysicalPackage'

DESC 'The PhysicalPackage class represents Physicalelements that contain or host other components. Examples are a Rack enclosure or an adapter Card.'

SUP dlm1Physicalelement ABSTRACT

MAY (\$ dlmDepth \$ dlmHeight \$ dlmHotSwappable \$ dlmRemovable \$ dlmReplaceable \$ dlmWeight \$ dlmWidth)

)

(1.3.6.1.4.1.412.100.3.1.2.2 NAME 'dlm1PhysicalPackageAuxClass'

DESC 'The PhysicalPackage class represents Physicalelements that contain or host other components. Examples are a Rack enclosure or an adapter Card.'

```

    SUP dlm1PhysicalPackage AUXILIARY
  )
  ( 1.3.6.1.4.1.412.100.3.1.2.3 NAME 'dlm1PhysicalPackageInstance'
    DESC 'The PhysicalPackage class represents
          PhysicalElements that contain or host other
          components. Examples are a Rack enclosure or an
          adapter Card.'
    SUP dlm1PhysicalPackage
  )
  ( 1.3.6.1.4.1.412.100.3.3.2.1 NAME
'dlm1PhysicalPackageInstanceNameForm'
    OC dlm1PhysicalPackageInstance
    MUST ( orderedCimKeys )
  )
  ( <physical-sr-1> NAME 'dlm1PhysicalPackageInstanceStructureRule'
    Form dlm1PhysicalPackageInstanceNameForm
  )

```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalPackageInstance.

```

  ( 1.3.6.1.4.1.412.100.3.1.2.3 NAME
'dlm1PhysicalPackageInstanceContentRule'
    DESC 'Aux classes that can attach to
          dlm1PhysicalPackageInstance.'
    AUX ( dlm1PackageInConnectorAuxClass $
          dlm1PackageInSlotAuxClass $
          dlm1ComputerSystemPackageHelperAuxClass $
          dlm1LibraryPackageAuxClass $
          dlm1PackageCoolingAuxClass $
          dlm1PackageTempSensorAuxClass $
          dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
          dlm1PhysicalElementLocationAuxClass $
          dlm1ElementCapacityAuxClass $
          dlm1ElementsLinkedAuxClass $
          dlm1ElementConfigurationAuxClass $
          dlm1ElementSettingAuxClass $
          dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
          dlm1StatisticsAuxClass $ dlm1PackageInChassisAuxClass $
          dlm1ProductPhysicalElementsAuxClass $
          dlm1FRUPhysicalElementsAuxClass $
          dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
          $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
          dlm1SystemComponentAuxClass $
          dlm1MemberOfCollectionAuxClass $
          dlm1PackagedComponentAuxClass $
          dlm1ConnectorOnPackageAuxClass )
  )

```

3.2 PhysicalFrame Classes

These classes function as the superclass for Rack, Chassis and other frame enclosures, which are defined as subclasses. As the superclass, it holds attributes related to alarms and security breaches.

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
( 1.3.6.1.4.1.412.100.3.2.1008 NAME 'dlmAudibleAlarm'
  DESC 'Boolean indicating whether the Frame is equipped
        with an audible alarm.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1009 NAME 'dlmBreachDescription'
  DESC 'BreachDescription is a free-form string providing
        more information if the SecurityBreach property
        indicates that a breach or some other security-related
        event occurred.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1010 NAME 'dlmCableManagementStrategy'
  DESC 'CableManagementStrategy is a free-form string
        that contains information on how the various cables
        are connected and bundled for the Frame. With many
        networking, storage-related and power cables, cable
        management can be a complex and challenging endeavor.
        This string property contains information to aid in
        assembly and service of the Frame.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1011 NAME 'dlmIsLocked'
  DESC 'Boolean indicating that the Frame is currently
        locked.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1012 NAME 'dlmLockPresent'
  DESC 'Boolean indicating whether the Frame is protected
        with a lock.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1013 NAME 'dlmSecurityBreach'
  DESC 'SecurityBreach is an enumerated, integer-valued
        property indicating whether a physical breach of the
        Frame was attempted but unsuccessful (value=4) or
        attempted and successful (5). Also, the values,
        "Unknown", "Other" or "No Breach", can be specified.
        Values are 1="Other", 2="Unknown", 3="No Breach",
        4="Breach Attempted", 5="Breach Successful".'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1016 NAME 'dlmVisibleAlarm'
  DESC 'Boolean indicating that the equipment includes a
        visible alarm.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
( 1.3.6.1.4.1.412.100.3.1.2.4 NAME 'dlm1PhysicalFrame'
  DESC 'PhysicalFrame is a superclass of Rack, Chassis
        and other frame enclosures, as they are defined in
        extension classes. Properties like visible or audible
        alarm, and data related to security breaches are in
        this superclass.'
  SUP dlm1PhysicalPackage ABSTRACT
  MAY ( dlmAudibleAlarm $ dlmBreachDescription $
        dlmCableManagementStrategy $ dlmIsLocked $
        dlmLockPresent $ dlmSecurityBreach $
        dlmVisibleAlarm )
)

( 1.3.6.1.4.1.412.100.3.1.2.5 NAME 'dlm1PhysicalFrameAuxClass'
  DESC 'PhysicalFrame is a superclass of Rack, Chassis
        and other frame enclosures, as they are defined in
        extension classes. Properties like visible or audible
        alarm, and data related to security breaches are in
        this superclass.'
  SUP dlm1PhysicalFrame AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.6 NAME 'dlm1PhysicalFrameInstance'
  DESC 'PhysicalFrame is a superclass of Rack, Chassis
        and other frame enclosures, as they are defined in
        extension classes. Properties like visible or audible
        alarm, and data related to security breaches are in
        this superclass.'
  SUP dlm1PhysicalFrame
)

( 1.3.6.1.4.1.412.100.3.3.2.2 NAME
'dlm1PhysicalFrameInstanceNameForm'
  OC dlm1PhysicalFrameInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-2> NAME 'dlm1PhysicalFrameInstanceStructureRule'
  Form dlm1PhysicalFrameInstanceNameForm
)
```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalFrameInstance.

```
( 1.3.6.1.4.1.412.100.3.1.2.6 NAME
'dlm1PhysicalFrameInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1PhysicalFrameInstance.'
  AUX ( dlm1PackageInConnectorAuxClass $
        dlm1PackageInSlotAuxClass $
        dlm1ComputerSystemPackageHelperAuxClass $
        dlm1LibraryPackageAuxClass $
        dlm1PackageCoolingAuxClass $
        dlm1PackageTempSensorAuxClass $
        dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
```

```

    dlm1ElementsLinkedAuxClass $
    dlm1ElementConfigurationAuxClass $
    dlm1ElementSettingAuxClass $
    dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
    dlm1ProvidesServiceToElementAuxClass $
    dlm1StatisticsAuxClass $ dlm1PackageInChassisAuxClass $
    dlm1ProductPhysicalElementsAuxClass $
    dlm1FRUPhysicalElementsAuxClass $
    dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
    $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
    dlm1SystemComponentAuxClass $
    dlm1MemberOfCollectionAuxClass $
    dlm1PackagedComponentAuxClass $
    dlm1ConnectorOnPackageAuxClass )
)

```

3.3 Rack Classes

These classes represent the enclosure in which Chassis are placed. Typically the Rack is nothing more than the enclosure, with functioning components being packaged in the Chassis that is loaded into the Rack.

```

( 1.3.6.1.4.1.412.100.3.2.1018 NAME 'dlmTypeOfRack'
  DESC 'Enumeration indicating the type of
    Rack. Information such as "Telco" rack (value=2) or
    standard 19 inch rack (1) can be specified. The
    country for which the Rack is manufactured is defined
    in the the Country Designation property. Values are
    0="Unknown", 1="Standard 19 Inch", 2="Telco",
    3="Equipment Shelf", 4="Non-Standard".'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.7 NAME 'dlm1Rack'
  DESC 'A Rack is a PhysicalFrame that represents an
    enclosure in which Chassis are placed. Typically a
    Rack is nothing more than the enclosure, and all the
    functioning componentry is packaged in the Chassis,
    loaded in the Rack.'
  SUP dlm1PhysicalFrame ABSTRACT
  MAY ( cc $ dlmHeight $ dlmTypeOfRack )
)

( 1.3.6.1.4.1.412.100.3.1.2.8 NAME 'dlm1RackAuxClass'
  DESC 'A Rack is a PhysicalFrame that represents an
    enclosure in which Chassis are placed. Typically a
    Rack is nothing more than the enclosure, and all the
    functioning componentry is packaged in the Chassis,
    loaded in the Rack.'
  SUP dlm1Rack AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.9 NAME 'dlm1RackInstance'
  DESC 'A Rack is a PhysicalFrame that represents an
    enclosure in which Chassis are placed. Typically a
    Rack is nothing more than the enclosure, and all the

```



```

        functioning componentry is packaged in the Chassis,
        loaded in the Rack.'
    SUP dlm1Rack
)

( 1.3.6.1.4.1.412.100.3.3.2.3 NAME 'dlm1RackInstanceNameForm'
  OC dlm1RackInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-3> NAME 'dlm1RackInstanceStructureRule'
  Form dlm1RackInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1RackInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.9 NAME 'dlm1RackInstanceContentRule'
  DESC 'Aux classes that can attach to dlm1RackInstance.'
  AUX ( dlm1PackageInConnectorAuxClass $
        dlm1PackageInSlotAuxClass $
        dlm1ComputerSystemPackageHelperAuxClass $
        dlm1LibraryPackageAuxClass $
        dlm1PackageCoolingAuxClass $
        dlm1PackageTempSensorAuxClass $
        dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1PackageInChassisAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass $
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass $
        dlm1ChassisInRackAuxClass $
        dlm1PackagedComponentAuxClass $
        dlm1ConnectorOnPackageAuxClass )
)

```

3.4 Chassis Classes

These classes represent the PhysicalElements that enclose other Elements and provide definable functionality, such as a desktop processing node, UPS, disk or tape storage, or some combination.

```

( 1.3.6.1.4.1.412.100.3.2.1020 NAME 'dlmCurrentRequiredOrProduced'
  DESC 'Current required by the Chassis at 120V. If power
        is provided by the Chassis (as in the case of a UPS),
        this property may indicate the amperage produced, as a
        negative number. The value is considered to be Amps at
        120 Volts.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
)

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1021 NAME 'dlmHeatGeneration'
  DESC 'Amount of heat generated by the Chassis in
        BTU/hour. The value is considered to be BTU per Hour.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1022 NAME 'dlmNumberOfPowerCords'
  DESC 'Integer indicating the number of power cords
        which must be connected to the Chassis, for all the
        componentry to operate.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.10 NAME 'dlm1Chassis'
  DESC 'The Chassis class represents the PhysicalElements
        that enclose other Elements and provide definable
        functionality, such as a desktop, processing node,
        UPS, disk or tape storage, or a combination of these.'
  SUP dlm1PhysicalFrame ABSTRACT
  MAY ( dlmCurrentRequiredOrProduced $ dlmHeatGeneration $
        dlmNumberOfPowerCords )
)

( 1.3.6.1.4.1.412.100.3.1.2.11 NAME 'dlm1ChassisAuxClass'
  DESC 'The Chassis class represents the PhysicalElements
        that enclose other Elements and provide definable
        functionality, such as a desktop, processing node,
        UPS, disk or tape storage, or a combination of these.'
  SUP dlm1Chassis AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.12 NAME 'dlm1ChassisInstance'
  DESC 'The Chassis class represents the PhysicalElements
        that enclose other Elements and provide definable
        functionality, such as a desktop, processing node,
        UPS, disk or tape storage, or a combination of these.'
  SUP dlm1Chassis
)

( 1.3.6.1.4.1.412.100.3.3.2.4 NAME 'dlm1ChassisInstanceNameForm'
  OC dlm1ChassisInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-4> NAME 'dlm1ChassisInstanceStructureRule'
  Form dlm1ChassisInstanceNameForm
)

The following content rule shows what auxiliary classes may be attached
to dlm1ChassisInstance.

( 1.3.6.1.4.1.412.100.3.1.2.12 NAME 'dlm1ChassisInstanceContentRule'
```

```

DESC 'Aux classes that can attach to
      dlm1ChassisInstance.'
AUX ( dlm1DockedAuxClass $
      dlm1PackageInConnectorAuxClass $
      dlm1PackageInSlotAuxClass $
      dlm1ComputerSystemPackageHelperAuxClass $
      dlm1LibraryPackageAuxClass $
      dlm1PackageCoolingAuxClass $
      dlm1PackageTempSensorAuxClass $
      dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
      dlm1PhysicalElementLocationAuxClass $
      dlm1ElementCapacityAuxClass $
      dlm1ElementsLinkedAuxClass $
      dlm1ElementConfigurationAuxClass $
      dlm1ElementSettingAuxClass $
      dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
      dlm1ProvidesServiceToElementAuxClass $
      dlm1StatisticsAuxClass $ dlm1ChassisInRackAuxClass $
      dlm1PackageInChassisAuxClass $
      dlm1ProductPhysicalElementsAuxClass $
      dlm1FRUPhysicalElementsAuxClass $
      dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
      $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
      dlm1SystemComponentAuxClass $
      dlm1MemberOfCollectionAuxClass $
      dlm1PackagedComponentAuxClass $
      dlm1ConnectorOnPackageAuxClass )
)

```

3.5 Card Classes

These classes represent the types of physical containers that can either be plugged into another Card or HostingBoard, or are themselves a HostingBoard/Motherboard in a Chassis. They include any package capability of carrying signals and providing a mounting point for PhysicalComponents (Chips) or other PhysicalPackages (Cards).

```

( 1.3.6.1.4.1.412.100.3.2.1024 NAME 'dlmHostingBoard'
  DESC 'Boolean indicating that this Card is a
        Motherboard or, more generically, a baseboard in a
        Chassis.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1025 NAME 'dlmOperatingVoltages'
  DESC 'Operating voltages required by the Card. The
        value is considered to be MilliVolts.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1026 NAME 'dlmRequirementsDescription'
  DESC 'A free-form string describing the way(s) in which
        this Card is physically unique from other Cards. This
        property only has meaning when the corresponding
        boolean property, SpecialRequirements, is set to TRUE.'
)

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1027 NAME 'dlmRequiresDaughterBoard'
DESC 'Boolean indicating that at least one
      daughterboard or auxiliary Card is required in order
      to function properly.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1028 NAME 'dlmSlotLayout'
DESC 'SlotLayout is a free-form string that describes
      the slot positioning, typical usage, restrictions,
      individual slot spacings or any other pertinent
      information for the slots on a Card.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1029 NAME 'dlmSpecialRequirements'
DESC 'Boolean indicating that this Card is physically
      unique from other Cards of the same type and therefore
      requires a special Slot. For example, a double-wide
      Card requires two Slots. Another example is where a
      certain Card may be used for the same general function
      as other Cards but requires a special Slot (e.g., extra
      long), whereas the other Cards can be placed in any
      available Slot. If set to TRUE, then the corresponding
      property, RequirementsDescription, should specify the
      nature of the uniqueness or purpose of the Card.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.1.2.13 NAME 'dlm1Card'
DESC 'The Card class represents a type of physical
      container that can be plugged into another Card or
      HostingBoard, or is itself a HostingBoard/Motherboard
      in a Chassis. The Card class includes any package
      capable of carrying signals and providing a mounting
      point for PhysicalComponents, such as Chips, or other
      PhysicalPackages, such as other Cards.'
SUP dlm1PhysicalPackage ABSTRACT
MAY ( dlmHostingBoard $ dlmOperatingVoltages $
      dlmRequirementsDescription $ dlmRequiresDaughterBoard
      $ dlmSlotLayout $ dlmSpecialRequirements )
)

( 1.3.6.1.4.1.412.100.3.1.2.14 NAME 'dlm1CardAuxClass'
DESC 'The Card class represents a type of physical
      container that can be plugged into another Card or
      HostingBoard, or is itself a HostingBoard/Motherboard
      in a Chassis. The Card class includes any package
      capable of carrying signals and providing a mounting
      point for PhysicalComponents, such as Chips, or other
      PhysicalPackages, such as other Cards.'
SUP dlm1Card AUXILIARY
```

```

)
( 1.3.6.1.4.1.412.100.3.1.2.15 NAME 'dlm1CardInstance'
  DESC 'The Card class represents a type of physical
        container that can be plugged into another Card or
        HostingBoard, or is itself a HostingBoard/Motherboard
        in a Chassis. The Card class includes any package
        capable of carrying signals and providing a mounting
        point for PhysicalComponents, such as Chips, or other
        PhysicalPackages, such as other Cards.'
  SUP dlm1Card
)
( 1.3.6.1.4.1.412.100.3.3.2.5 NAME 'dlm1CardInstanceNameForm'
  OC dlm1CardInstance
  MUST ( orderedCimKeys )
)
( <physical-sr-5> NAME 'dlm1CardInstanceStructureRule'
  Form dlm1CardInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1CardInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.15 NAME 'dlm1CardInstanceContentRule'
  DESC 'Aux classes that can attach to dlm1CardInstance.'
  AUX ( dlm1CardInSlotAuxClass $
        dlm1PackageInConnectorAuxClass $
        dlm1PackageInSlotAuxClass $
        dlm1ComputerSystemPackageHelperAuxClass $
        dlm1LibraryPackageAuxClass $
        dlm1PackageCoolingAuxClass $
        dlm1PackageTempSensorAuxClass $
        dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1CardOnCardAuxClass $
        dlm1PackageInChassisAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass $
        dlm1MemoryOnCardAuxClass $
        dlm1PackagedComponentAuxClass $
        dlm1ConnectorOnPackageAuxClass )
)

```

3.6 SystemBusCard Classes

These classes provide addition information, such as bus type and data width, which dictate what type of Slot the Card can be inserted into. It is possible, with this class to define a Card as a 64 bit, PCI adapter.

```
( 1.3.6.1.4.1.412.100.3.2.1030 NAME 'dlmBusType'
DESC 'An enumerated integer describing the System bus
type for this Card. It indicates the type of Slot into
which the Card can plug. The list of permissible values
aligns with the System bus types in
PhysicalConnector.Connector Type. Values are
43="PCI", 44="ISA", 45="EISA", 46="VESA", 47="PCMCIA",
48="PCMCIA Type I", 49="PCMCIA Type II", 50="PCMCIA
Type III", 52="CardBus", 64="Access.bus", 65="NuBus",
73="AGP", 74="VME Bus", 75="VME64", 76="Proprietary",
77="Proprietary Processor Card Slot", 78="Proprietary
Memory Card Slot", 79="Proprietary I/O Riser Slot",
80="PCI-66MHZ", 81="AGP2X", 82="AGP4X", 83="PC-98",
84="PC-98-Hireso", 85="PC-H98", 86="PC-98Note",
87="PC-98Full", 98="PCI-X", 99="Sbus IEEE 1396-1993 32
bit", 100="Sbus IEEE 1396-1993 64 bit", 101="MCA",
102="GIO", 103="XIO", 104="HIO", 105="NGIO",
106="PMC", 109="Future I/O", 110="InfiniBand".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1031 NAME 'dlmBusWidth'
DESC 'System bus width (in bits) required by this Card.
If "unknown", enter 0. If "other" than the values, 8,
16, 32, 64 or 128, enter 1. The list of permissible
values aligns with the data in Slot.MaxBusWidth.
Value Mappings are "0", "1", "8", "16", "32", "64",
"128" The value is considered to be Bits.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.16 NAME 'dlm1SystemBusCard'
DESC 'The SystemBusCard class represents additional
information for a Card, detailing the Card's bus type
and data width. These properties dictate the type of
Slot into which the Card can be inserted. For example,
using the properties of this class, one can define that
a Card is a PCI, 64 bit adapter.'
SUP dlm1Card ABSTRACT
MAY ( dlmBusType $ dlmBusWidth )
)

( 1.3.6.1.4.1.412.100.3.1.2.17 NAME 'dlm1SystemBusCardAuxClass'
DESC 'The SystemBusCard class represents additional
information for a Card, detailing the Card's bus type
and data width. These properties dictate the type of
Slot into which the Card can be inserted. For example,
using the properties of this class, one can define that
a Card is a PCI, 64 bit adapter.'
SUP dlm1SystemBusCard AUXILIARY
```

```

)
( 1.3.6.1.4.1.412.100.3.1.2.18 NAME 'dlm1SystemBusCardInstance'
  DESC 'The SystemBusCard class represents additional
        information for a Card, detailing the Card"s bus type
        and data width. These properties dictate the type of
        Slot into which the Card can be inserted. For example,
        using the properties of this class, one can define that
        a Card is a PCI, 64 bit adapter.'
  SUP dlm1SystemBusCard
)
( 1.3.6.1.4.1.412.100.3.3.2.6 NAME
'dlm1SystemBusCardInstanceNameForm'
  OC dlm1SystemBusCardInstance
  MUST ( orderedCimKeys )
)
( <physical-sr-6> NAME 'dlm1SystemBusCardInstanceStructureRule'
  Form dlm1SystemBusCardInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1SystemBusCardInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.18 NAME
'dlm1SystemBusCardInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1SystemBusCardInstance.'
  AUX ( dlm1CardInSlotAuxClass $
        dlm1PackageInConnectorAuxClass $
        dlm1PackageInSlotAuxClass $
        dlm1ComputerSystemPackageHelperAuxClass $
        dlm1LibraryPackageAuxClass $
        dlm1PackageCoolingAuxClass $
        dlm1PackageTempSensorAuxClass $
        dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1CardOnCardAuxClass $
        dlm1PackageInChassisAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass $
        dlm1MemoryOnCardAuxClass $
        dlm1PackagedComponentAuxClass $
        dlm1ConnectorOnPackageAuxClass )
)

```

3.7 StorageMediaLocation Classes

These classes model PhysicalElements where PhysicalMedia may be placed. They describe an entity that holds Media, rather than a "place," which would be conveyed by a Location entry. Typically, these classes are used in the context of a StorageLibrary. Examples are MediaAccessDevices, InterLibraryPorts or "slots" in the Library's panel.

```
( 1.3.6.1.4.1.412.100.3.2.1032 NAME 'dlmLocationCoordinates'
DESC 'LocationCoordinates represent the physical
      location of the the StorageMediaLocation instance. The
      property is defined as a free-form string to allow the
      location information to be described in vendor-unique
      terminology.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1033 NAME 'dlmLocationType'
DESC 'The type of Location. For example, whether this
      is an individual Media "Slot" (value=2), a
      MediaAccessDevice (value=4) or a "Magazine" (value=3)
      is indicated in this property. Values are
      0="Unknown", 1="Other", 2="Slot", 3="Magazine",
      4="MediaAccessDevice", 5="InterLibrary Port",
      6="Limited Access Port", 7="Door", 8="Shelf",
      9="Vault".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1034 NAME 'dlmMediaCapacity'
DESC 'A StorageMediaLocation may hold more than one
      PhysicalMedia - for example, a Magazine. This property
      indicates the Physical Media capacity of the Location.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.19 NAME 'dlm1StorageMediaLocation'
DESC 'StorageMediaLocation is a PhysicalElement where
      PhysicalMedia may be placed. This class describes an
      entity that holds Media and is not just a "place" (as
      is conveyed by the Location object). This class is
      typically used in the context of a StorageLibrary.
      Examples of StorageMediaLocations are Media
      AccessDevices, InterLibraryPorts or "slots" in a
      Library's panel.'
SUP dlm1PhysicalPackage ABSTRACT
MAY ( dlmLocationCoordinates $ dlmLocationType $
      dlmMediaCapacity $ dlmMediaSizesSupported $
      dlmMediaTypesSupported $ dlmTypeDescriptions )
)

( 1.3.6.1.4.1.412.100.3.1.2.20 NAME
'dlm1StorageMediaLocationAuxClass'
DESC 'StorageMediaLocation is a PhysicalElement where
      PhysicalMedia may be placed. This class describes an
```



```

        entity that holds Media and is not just a "place" (as
        is conveyed by the Location object). This class is
        typically used in the context of a StorageLibrary.
        Examples of StorageMediaLocations are Media
        AccessDevices, InterLibraryPorts or "slots" in a
        Library's panel.'
    SUP dlm1StorageMediaLocation AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.21 NAME
'dlm1StorageMediaLocationInstance'
  DESC 'StorageMediaLocation is a PhysicalElement where
  PhysicalMedia may be placed. This class describes an
  entity that holds Media and is not just a "place" (as
  is conveyed by the Location object). This class is
  typically used in the context of a StorageLibrary.
  Examples of StorageMediaLocations are Media
  AccessDevices, InterLibraryPorts or "slots" in a
  Library's panel.'
  SUP dlm1StorageMediaLocation
)

( 1.3.6.1.4.1.412.100.3.3.2.7 NAME
'dlm1StorageMediaLocationInstanceNameForm'
  OC dlm1StorageMediaLocationInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-7> NAME
'dlm1StorageMediaLocationInstanceStructureRule'
  Form dlm1StorageMediaLocationInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1StorageMediaLocationInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.21 NAME
'dlm1StorageMediaLocationInstanceContentRule'
  DESC 'Aux classes that can attach to
  dlm1StorageMediaLocationInstance.'
  AUX ( dlm1DeviceServicesLocationHelperAuxClass $
  dlm1PhysicalMediaInLocationAuxClass $
  dlm1HomeForMediaAuxClass $
  dlm1PackageInConnectorAuxClass $
  dlm1PackageInSlotAuxClass $
  dlm1ComputerSystemPackageHelperAuxClass $
  dlm1LibraryPackageAuxClass $
  dlm1PackageCoolingAuxClass $
  dlm1PackageTempSensorAuxClass $
  dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
  dlm1PhysicalElementLocationAuxClass $
  dlm1ElementCapacityAuxClass $
  dlm1ElementsLinkedAuxClass $
  dlm1ElementConfigurationAuxClass $
  dlm1ElementSettingAuxClass $
  dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
  dlm1ProvidesServiceToElementAuxClass $

```

```

    dlm1StatisticsAuxClass $ dlm1PackageInChassisAuxClass $
    dlm1ProductPhysicalElementsAuxClass $
    dlm1FRUPhysicalElementsAuxClass $
    dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
    $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
    dlm1SystemComponentAuxClass $
    dlm1MemberOfCollectionAuxClass $
    dlm1PackagedComponentAuxClass $
    dlm1ConnectorOnPackageAuxClass )
)

```

3.8 Magazine Classes

While StorageMediaLocations are typically slots or spaces where removable media are located, these classes represent a single physical container with multiple StorageMediaLocations. All the MediaLocations within a Magazine are added/removed together, often with some label for identification (the unique data captured in the Magazine class). Label properties are defined the same way as labels for PhysicalMedia.

```

( 1.3.6.1.4.1.412.100.3.1.2.22 NAME 'dlm1Magazine'
  DESC 'StorageMediaLocations are typically slots or
        spaces where removable media are located. However, a
        specific kind of MediaLocation is a Magazine. This
        entity represents a single physical container with
        multiple StorageMediaLocations in it. All the
        MediaLocations within the Magazine are added/removed
        together. Often, this container has a Barcode or other
        label for identification. This is the unique data
        captured in the Magazine class. Magazine"s label
        properties are defined exactly as are labels for
        PhysicalMedia. '
  SUP dlm1StorageMediaLocation ABSTRACT
)

( 1.3.6.1.4.1.412.100.3.1.2.23 NAME 'dlm1MagazineAuxClass'
  DESC 'StorageMediaLocations are typically slots or
        spaces where removable media are located. However, a
        specific kind of MediaLocation is a Magazine. This
        entity represents a single physical container with
        multiple StorageMediaLocations in it. All the
        MediaLocations within the Magazine are added/removed
        together. Often, this container has a Barcode or other
        label for identification. This is the unique data
        captured in the Magazine class. Magazine"s label
        properties are defined exactly as are labels for
        PhysicalMedia. '
  SUP dlm1Magazine AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.24 NAME 'dlm1MagazineInstance'
  DESC 'StorageMediaLocations are typically slots or
        spaces where removable media are located. However, a
        specific kind of MediaLocation is a Magazine. This
        entity represents a single physical container with
        multiple StorageMediaLocations in it. All the

```

MediaLocations within the Magazine are added/removed together. Often, this container has a Barcode or other label for identification. This is the unique data captured in the Magazine class. Magazine's label properties are defined exactly as are labels for PhysicalMedia. '

```

SUP dlm1Magazine
)
( 1.3.6.1.4.1.412.100.3.3.2.8 NAME 'dlm1MagazineInstanceNameForm'
  OC dlm1MagazineInstance
  MUST ( orderedCimKeys )
)
( <physical-sr-8> NAME 'dlm1MagazineInstanceStructureRule'
  Form dlm1MagazineInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1MagazineInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.24 NAME 'dlm1MagazineInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1MagazineInstance.'
  AUX ( dlm1DeviceServicesLocationHelperAuxClass $
        dlm1PhysicalMediaInLocationAuxClass $
        dlm1HomeForMediaAuxClass $
        dlm1PackageInConnectorAuxClass $
        dlm1PackageInSlotAuxClass $
        dlm1ComputerSystemPackageHelperAuxClass $
        dlm1LibraryPackageAuxClass $
        dlm1PackageCoolingAuxClass $
        dlm1PackageTempSensorAuxClass $
        dlm1PackageAlarmAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1PackageInChassisAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass $
        dlm1PackagedComponentAuxClass $
        dlm1ConnectorOnPackageAuxClass )
)

```

3.9 PhysicalComponent Classes

These classes represent low-level/basic Components within a Package. These Components either cannot or do not need to be decomposed. For example, an ASIC (or Chip), or a tape for data storage (PhysicalMedia)

do not need to be decomposed. Links, Connectors and Packages are not descended from this class, but all others are.

```
( 1.3.6.1.4.1.412.100.3.1.2.25 NAME 'dlm1PhysicalComponent'
DESC 'The PhysicalComponent class represents any
low-level or basic Component within a Package. A
Component object either can not or does not need to be
decomposed into its constituent parts. For example, an
ASIC (or Chip) can not be further decomposed. A tape
for data storage (PhysicalMedia) does not need to be
decomposed. Any PhysicalElement that is not a Link,
Connector, or Package is a descendent (or member) of
this class. For example, the UART chipset on an
internal modem Card would be a subclass (if additional
properties or associations are defined) or an instance
of PhysicalComponent.'
SUP dlm1PhysicalElement ABSTRACT
MAY ( dlmHotSwappable $ dlmRemovable $ dlmReplaceable )
)

( 1.3.6.1.4.1.412.100.3.1.2.26 NAME 'dlm1PhysicalComponentAuxClass'
DESC 'The PhysicalComponent class represents any
low-level or basic Component within a Package. A
Component object either can not or does not need to be
decomposed into its constituent parts. For example, an
ASIC (or Chip) can not be further decomposed. A tape
for data storage (PhysicalMedia) does not need to be
decomposed. Any PhysicalElement that is not a Link,
Connector, or Package is a descendent (or member) of
this class. For example, the UART chipset on an
internal modem Card would be a subclass (if additional
properties or associations are defined) or an instance
of PhysicalComponent.'
SUP dlm1PhysicalComponent AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.27 NAME 'dlm1PhysicalComponentInstance'
DESC 'The PhysicalComponent class represents any
low-level or basic Component within a Package. A
Component object either can not or does not need to be
decomposed into its constituent parts. For example, an
ASIC (or Chip) can not be further decomposed. A tape
for data storage (PhysicalMedia) does not need to be
decomposed. Any PhysicalElement that is not a Link,
Connector, or Package is a descendent (or member) of
this class. For example, the UART chipset on an
internal modem Card would be a subclass (if additional
properties or associations are defined) or an instance
of PhysicalComponent.'
SUP dlm1PhysicalComponent
)

( 1.3.6.1.4.1.412.100.3.3.2.9 NAME
'dlm1PhysicalComponentInstanceNameForm'
OC dlm1PhysicalComponentInstance
MUST ( orderedCimKeys )
)
```

```
( <physical-sr-9> NAME 'dlm1PhysicalComponentInstanceStructureRule'
  Form dlm1PhysicalComponentInstanceNameForm
)
```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalComponentInstance.

```
( 1.3.6.1.4.1.412.100.3.1.2.27 NAME
'dlm1PhysicalComponentInstanceContentRule'
  DESC 'Aux classes that can attach to
  dlm1PhysicalComponentInstance.'
  AUX ( dlm1RealizesExtentAuxClass $ dlm1RealizesAuxClass
    $ dlm1PhysicalElementLocationAuxClass $
    dlm1ElementCapacityAuxClass $
    dlm1ElementsLinkedAuxClass $
    dlm1ElementConfigurationAuxClass $
    dlm1ElementSettingAuxClass $
    dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
    dlm1ProvidesServiceToElementAuxClass $
    dlm1StatisticsAuxClass $ dlm1PackagedComponentAuxClass $
    dlm1ProductPhysicalElementsAuxClass $
    dlm1FRUPhysicalElementsAuxClass $
    dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
    $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
    dlm1SystemComponentAuxClass $
    dlm1MemberOfCollectionAuxClass )
)
```

3.10 Chip Classes

These classes represents any type of integrated circuit hardware, including ASICs, processors, memory chips, etc.

```
( 1.3.6.1.4.1.412.100.3.2.1040 NAME 'dlmFormFactor'
  DESC 'The implementation form factor for the Chip.For
  example, values such as SIMM (7), TSOP (9) or PGA (10)
  can be specified. Values are 0="Unknown", 1="Other",
  2="SIP", 3="DIP", 4="ZIP", 5="SOJ", 6="Proprietary",
  7="SIMM", 8="DIMM", 9="TSOP", 10="PGA", 11="RIMM",
  12="SODIMM", 13="SRIMM", 14="SMD", 15="SSMP",
  16="QFP", 17="TQFP", 18="SOIC", 19="LCC", 20="PLCC",
  21="BGA", 22="FPBGA", 23="LGA".'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.28 NAME 'dlm1Chip'
  DESC 'The Chip class represents any type of integrated
  circuit hardware, including ASICs, processors, memory
  chips, etc.'
  SUP dlm1PhysicalComponent ABSTRACT
  MAY ( dlmFormFactor )
)

( 1.3.6.1.4.1.412.100.3.1.2.29 NAME 'dlm1ChipAuxClass'
  DESC 'The Chip class represents any type of integrated
  circuit hardware, including ASICs, processors, memory
```

```

        chips, etc.'
    SUP dlm1Chip AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.30 NAME 'dlm1ChipInstance'
  DESC 'The Chip class represents any type of integrated
        circuit hardware, including ASICs, processors, memory
        chips, etc.'
  SUP dlm1Chip
)

( 1.3.6.1.4.1.412.100.3.3.2.10 NAME 'dlm1ChipInstanceNameForm'
  OC dlm1ChipInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-10> NAME 'dlm1ChipInstanceStructureRule'
  Form dlm1ChipInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1ChipInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.30 NAME 'dlm1ChipInstanceContentRule'
  DESC 'Aux classes that can attach to dlm1ChipInstance.'
  AUX ( dlm1RealizesExtentAuxClass $ dlm1RealizesAuxClass
        $ dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1PackagedComponentAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass )
)

```

3.11 PhysicalMemory Classes

These classes represent low level memory devices, like SIMMs, DIMMs, and raw memory chips.

```

( 1.3.6.1.4.1.412.100.3.2.1041 NAME 'dlmBankLabel'
  DESC 'A string identifying the physically labeled bank
        where the Memory is located - for example, "Bank 0" or
        "Bank A".'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{64} SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1042 NAME 'dlmCapacity'
  DESC 'The total capacity of this PhysicalMemory, in
        bytes. The value is considered to be Bytes.'
)

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1043 NAME 'dlmDataWidth'
DESC 'Data width of the PhysicalMemory, in bits. A data
width of 0 and a TotalWidth of 8 would indicate that
the Memory is solely used to provide error correction
bits. The value is considered to be Bits.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1044 NAME 'dlmInterleavePosition'
DESC 'The position of this PhysicalMemory in an
interleave. 0 indicates non-interleaved. 1 indicates
the first position, 2 the second position and so on.
For example, in a 2:1 interleave, a value of "1" would
indicate that the Memory is in the "even" position.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1045 NAME 'dlmMemoryType'
DESC 'The type of PhysicalMemory. Values are
0="Unknown", 1="Other", 2="DRAM", 3="Synchronous
DRAM", 4="Cache DRAM", 5="EDO", 6="EDRAM", 7="VRAM",
8="SRAM", 9="RAM", 10="ROM", 11="Flash", 12="EEPROM",
13="FEPROM", 14="EPROM", 15="CDRAM", 16="3DRAM",
17="SDRAM", 18="SGRAM", 19="RDRAM", 20="DDR".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1046 NAME 'dlmPositionInRow'
DESC 'Specifies the position of the PhysicalMemory in a
"row". For example, if it takes two 8-bit memory
devices to form a 16bit row, then a value of "2" means
that this Memory is the second device. 0 is an invalid
value for this property.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1047 NAME 'dlmSpeed'
DESC 'The speed of the PhysicalMemory, in nanoseconds.
The value is considered to be NanoSeconds.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1048 NAME 'dlmTotalWidth'
DESC 'Total width, in bits, of the PhysicalMemory,
including check or error correction bits. If there are
no error correction bits, the value in this property
should match that specified for DataWidth. The value
is considered to be Bits.'
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.31 NAME 'dlm1PhysicalMemory'
DESC 'PhysicalMemory is a subclass of Chip,
      representing low level memory devices - SIMMS, DIMMs,
      raw memory chips, etc.'
SUP dlm1Chip ABSTRACT
MAY ( dlmBankLabel $ dlmCapacity $ dlmDataWidth $
      dlmFormFactor $ dlmInterleavePosition $ dlmMemoryType
      $ dlmPositionInRow $ dlmSpeed $ dlmTotalWidth )
)

( 1.3.6.1.4.1.412.100.3.1.2.32 NAME 'dlm1PhysicalMemoryAuxClass'
DESC 'PhysicalMemory is a subclass of Chip,
      representing low level memory devices - SIMMS, DIMMs,
      raw memory chips, etc.'
SUP dlm1PhysicalMemory AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.33 NAME 'dlm1PhysicalMemoryInstance'
DESC 'PhysicalMemory is a subclass of Chip,
      representing low level memory devices - SIMMS, DIMMs,
      raw memory chips, etc.'
SUP dlm1PhysicalMemory
)

( 1.3.6.1.4.1.412.100.3.3.2.11 NAME
'dlm1PhysicalMemoryInstanceNameForm'
OC dlm1PhysicalMemoryInstance
MUST ( orderedCimKeys )
)

( <physical-sr-11> NAME 'dlm1PhysicalMemoryInstanceStructureRule'
Form dlm1PhysicalMemoryInstanceNameForm
)
```

The following content rule shows what auxiliary classes may be attached to `dlm1PhysicalMemoryInstance`.

```
( 1.3.6.1.4.1.412.100.3.1.2.33 NAME
'dlm1PhysicalMemoryInstanceContentRule'
DESC 'Aux classes that can attach to
      dlm1PhysicalMemoryInstance.'
AUX ( dlm1MemoryWithMediaAuxClass $
      dlm1RealizesExtentAuxClass $ dlm1RealizesAuxClass $
      dlm1PhysicalElementLocationAuxClass $
      dlm1ElementCapacityAuxClass $
      dlm1ElementsLinkedAuxClass $
      dlm1ElementConfigurationAuxClass $
      dlm1ElementSettingAuxClass $
      dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
      dlm1ProvidesServiceToElementAuxClass $
      dlm1StatisticsAuxClass $ dlm1MemoryOnCardAuxClass $
      dlm1PackagedComponentAuxClass $
      dlm1ProductPhysicalElementsAuxClass $
```



```

    dlm1FRUPhysicalElementsAuxClass $
    dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
    $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
    dlm1SystemComponentAuxClass $
    dlm1MemberOfCollectionAuxClass )
)

```

3.12 PhysicalMedia Classes

These classes represent documentation or storage media, such as tapes, CDROMs, etc. They are typically used to locate and manage RemovableMedia (as opposed to Media sealed with the MediaAccessDevice as a single package, e.g. hard disks). However "sealed" Media can also be modeled with these classes, where the PackagedComponent relationship is used to associate the Media with the PhysicalPackage..

```

( 1.3.6.1.4.1.412.100.3.2.1049 NAME 'dlmCleanerMedia'
  DESC 'Boolean indicating that the PhysicalMedia is used
        for cleaning purposes and not data storage.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1050 NAME 'dlmDualSided'
  DESC 'Boolean indicating that the Media has two
        recording sides (TRUE) or only a single side (FALSE).
        Examples of dual sided Media include DVD-ROM and some
        optical disks. Examples of single sided Media are
        tapes and CD-ROM.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1051 NAME 'dlmMaxMounts'
  DESC 'For removable Media, the maximum number of times
        that the Media can be mounted before it should be
        retired. For cleaner Media, this is the maximum number
        of Drive cleans that can be performed. For nonremovable
        Media, such as hard disks, this property is not
        applicable and should be set to 0.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1052 NAME 'dlmMediaDescription'
  DESC 'Additional detail related to the MediaType
        enumeration. For example, if value 3 ("QIC Cartridge\.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1053 NAME 'dlmMediaSize'
  DESC 'Size of the Media in inches. For example, "3.5"
        would be entered for a 3.5 inch disk, or "12" would be
        entered for a 12 inch optical disk. On the other hand,
        "0.5" would be defined for a 1/2 inch tape. The value
        is considered to be Inches.'

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SUP cimFloat32 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1054 NAME 'dlnMediaType'
  DESC 'Specifies the type of the PhysicalMedia, as an
        enumerated integer. The MediaDescription property is
        used to provide more explicit definition of the Media
        type, whether it is pre-formatted, compatability
        features, etc. Values are 0="Unknown", 1="Other",
        2="Tape Cartridge", 3="QIC Cartridge", 4="AIT
        Cartridge", 5="DTF Cartridge", 6="DAT Cartridge",
        7="8mm Tape Cartridge", 8="19mm Tape Cartridge",
        9="DLT Cartridge", 10="Half-Inch Magnetic Tape
        Cartridge", 11="Cartridge Disk", 12="JAZ Disk",
        13="ZIP Disk", 14="SyQuest Disk", 15="Winchester
        Removable Disk", 16="CD-ROM", 17="CD-ROM/XA",
        18="CD-I", 19="CD Recordable", 20="WORM",
        21="Magneto-Optical", 22="DVD", 23="DVD-RW+",
        24="DVD-RAM", 25="DVD-ROM", 26="DVD-Video", 27="Divx",
        28="Floppy/Diskette", 29="Hard Disk", 30="Memory Card",
        31="Hard Copy", 32="Clik Disk", 33="CD-RW", 34="CD-DA",
        35="CD+", 36="DVD Recordable", 37="DVD-RW",
        38="DVD-Audio", 39="DVD-5", 40="DVD-9", 41="DVD-10",
        42="DVD-18", 43="Magneto-Optical Rewriteable",
        44="Magneto-Optical Write Once", 45="Magneto-Optical
        Rewriteable (LIMDOW)", 46="Phase Change Write Once",
        47="Phase Change Rewriteable", 48="Phase Change Dual
        Rewriteable", 49="Ablative Write Once", 50="Near Field
        Recording", 51="MiniQic", 52="Travan", 53="8mm Metal
        Particle", 54="8mm Advanced Metal Evaporate",
        55="NCTP", 56="LTO Ultrium", 57="LTO Accelis", 58="9
        Track Tape", 59="18 Track Tape", 60="36 Track Tape",
        61="Magstar 3590", 62="Magstar MP", 63="D2 Tape",
        64="Tape - DST Small ", 65="Tape - DST Medium",
        66="Tape - DST Large".'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1055 NAME 'dlnMountCount'
  DESC 'For removable or cleaner Media, the number of
        times that the Media has been mounted for data
        transfer or to clean a Drive. For nonremovable Media,
        such as hard disks, this property is not applicable
        and should be set to 0.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1056 NAME 'dlnTimeOfLastMount'
  DESC 'For removable or cleaner Media, the date and time
        that the Media was last mounted. For nonremovable
        Media, such as hard disks, this property has no
        meaning and is not applicable.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 SINGLE-VALUE
  EQUALITY generalizedTimeMatch
)
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
( 1.3.6.1.4.1.412.100.3.2.1057 NAME 'dlmTotalMountTime'
  DESC 'For removable or cleaner Media, the total time
        (in seconds) that the Media has been mounted for data
        transfer or to clean a Drive. For nonremovable Media,
        such as hard disks, this property is not applicable
        and should be set to 0.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1058 NAME 'dlmWriteProtectOn'
  DESC 'Boolean specifying whether the Media is currently
        write protected by some kind of physical mechanism,
        such as a protect tab on a floppy diskette.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.1.2.34 NAME 'dlm1PhysicalMedia'
  DESC 'The PhysicalMedia class represents any type of
        documentation or storage medium, such as tapes,
        CDROMs, etc. This class is typically used to locate
        and manage Removable Media (versus Media sealed with
        the MediaAccessDevice, as a single Package, as is the
        case with hard disks). However, "sealed" Media can
        also be modeled using this class, where the Media
        would then be associated with the PhysicalPackage
        using the PackagedComponent relationship.'
  SUP dlm1PhysicalComponent ABSTRACT
  MAY ( dlmCapacity $ dlmCleanerMedia $ dlmDualSided $
        dlmMaxMounts $ dlmMountCount $ dlmTimeOfLastMount $
        dlmMediaDescription $ dlmMediaSize $ dlmMediaType $
        dlmTotalMountTime $ dlmWriteProtectOn )
)

( 1.3.6.1.4.1.412.100.3.1.2.35 NAME 'dlm1PhysicalMediaAuxClass'
  DESC 'The PhysicalMedia class represents any type of
        documentation or storage medium, such as tapes,
        CDROMs, etc. This class is typically used to locate
        and manage Removable Media (versus Media sealed with
        the MediaAccessDevice, as a single Package, as is the
        case with hard disks). However, "sealed" Media can
        also be modeled using this class, where the Media
        would then be associated with the PhysicalPackage
        using the PackagedComponent relationship.'
  SUP dlm1PhysicalMedia AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.36 NAME 'dlm1PhysicalMediaInstance'
  DESC 'The PhysicalMedia class represents any type of
        documentation or storage medium, such as tapes,
        CDROMs, etc. This class is typically used to locate
        and manage Removable Media (versus Media sealed with
        the MediaAccessDevice, as a single Package, as is the
        case with hard disks). However, "sealed" Media can
        also be modeled using this class, where the Media
        would then be associated with the PhysicalPackage
```

```

        using the PackagedComponent relationship.'
    SUP dlm1PhysicalMedia
)

( 1.3.6.1.4.1.412.100.3.3.2.12 NAME
'dlm1PhysicalMediaInstanceNameForm'
  OC dlm1PhysicalMediaInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-12> NAME 'dlm1PhysicalMediaInstanceStructureRule'
  Form dlm1PhysicalMediaInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalMediaInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.36 NAME
'dlm1PhysicalMediaInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1PhysicalMediaInstance.'
  AUX ( dlm1MemoryWithMediaAuxClass $
        dlm1PhysicalMediaInLocationAuxClass $
        dlm1RealizesPExtentAuxClass $
        dlm1RealizesDiskPartitionAuxClass $
        dlm1RealizesAggregatePExtentAuxClass $
        dlm1RealizedOnSideAuxClass $ dlm1HomeForMediaAuxClass
        $ dlm1RealizesExtentAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1PackagedComponentAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass )
)

```

3.13 PhysicalTape Classes

These classes represent additional Tape Media data, including tape length and whether it must be unloaded from BOT.

```

( 1.3.6.1.4.1.412.100.3.2.1059 NAME 'dlmTapeLength'
  DESC 'The physical length of the Tape in feet. The
        value is considered to be Feet.'
  SUP cimFloat32 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1060 NAME 'dlmUnloadAnywhere'
  DESC 'Boolean set to TRUE if the Tape can be unloaded
        at any position on the Media. It is set to FALSE if

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
        the tape must be at a certain position for unload -
        such as at the beginning of tape (BOT) area, or at
        mid-tape point for TapeDrives with mid-tape load.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.1.2.37 NAME 'dlm1PhysicalTape'
  DESC 'The PhysicalTape class represents additional data
        for a Tape Media. Information on the tape length and
        whether it must be unloaded from BOT are properties of
        this class.'
  SUP dlm1PhysicalMedia ABSTRACT
  MAY ( dlmTapeLength $ dlmUnloadAnywhere )
)

( 1.3.6.1.4.1.412.100.3.1.2.38 NAME 'dlm1PhysicalTapeAuxClass'
  DESC 'The PhysicalTape class represents additional data
        for a Tape Media. Information on the tape length and
        whether it must be unloaded from BOT are properties of
        this class.'
  SUP dlm1PhysicalTape AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.39 NAME 'dlm1PhysicalTapeInstance'
  DESC 'The PhysicalTape class represents additional data
        for a Tape Media. Information on the tape length and
        whether it must be unloaded from BOT are properties of
        this class.'
  SUP dlm1PhysicalTape
)

( 1.3.6.1.4.1.412.100.3.3.2.13 NAME
'dlm1PhysicalTapeInstanceNameForm'
  OC dlm1PhysicalTapeInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-13> NAME 'dlm1PhysicalTapeInstanceStructureRule'
  Form dlm1PhysicalTapeInstanceNameForm
)
```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalTapeInstance.

```
( 1.3.6.1.4.1.412.100.3.1.2.39 NAME
'dlm1PhysicalTapeInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1PhysicalTapeInstance.'
  AUX ( dlm1RealizesTapePartitionAuxClass $
        dlm1MemoryWithMediaAuxClass $
        dlm1PhysicalMediaInLocationAuxClass $
        dlm1RealizesPEExtentAuxClass $
        dlm1RealizesDiskPartitionAuxClass $
        dlm1RealizesAggregatePEExtentAuxClass $
        dlm1RealizedOnSideAuxClass $ dlm1HomeForMediaAuxClass
        $ dlm1RealizesExtentAuxClass $ dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
```

```

    dlm1ElementCapacityAuxClass $
    dlm1ElementsLinkedAuxClass $
    dlm1ElementConfigurationAuxClass $
    dlm1ElementSettingAuxClass $
    dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
    dlm1ProvidesServiceToElementAuxClass $
    dlm1StatisticsAuxClass $ dlm1PackagedComponentAuxClass $
    dlm1ProductPhysicalElementsAuxClass $
    dlm1FRUPhysicalElementsAuxClass $
    dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
    $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
    dlm1SystemComponentAuxClass $
    dlm1MemberOfCollectionAuxClass )
)

```

3.14 PhysicalLink Classes

These classes represent the cabling of PhysicalElements together, which can include serial, Ethernet or infrared Links. In many cases the numerous physical cables within a PhysicalPackage or Network will not be modeled, but where these cables or Links are critical components, or are tagged assets, these classes can be instantiated.

```

( 1.3.6.1.4.1.412.100.3.2.1061 NAME 'dlmLength'
  DESC 'The current length of the PhysicalLink in feet.
        For some connections, especially wireless
        technologies, this property may not be applicable and
        should be left uninitialized. The value is considered
        to be Feet.'
  SUP cimFloat64 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1062 NAME 'dlmMaxLength'
  DESC 'The maximum length of the PhysicalLink in feet.
        The value is considered to be Feet.'
  SUP cimFloat64 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1063 NAME 'dlmWired'
  DESC 'Boolean indicating whether the PhysicalLink is an
        actual cable (TRUE) or a wireless connection (FALSE).'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.1.2.40 NAME 'dlm1PhysicalLink'
  DESC 'The PhysicalLink class represents the cabling of
        PhysicalElements together. For example, serial or
        Ethernet cables, and infrared Links would be
        subclasses (if additional properties or associations
        are defined) or instances of PhysicalLink. In many
        cases, the numerous physical cables within a
        PhysicalPackage or Network will not be modeled.
        However, where these cables or Links are critical
        components, or are tagged assets of the company, these
        objects can be instantiated using this class or one of
        its descendent classes.'
)

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SUP dlm1PhysicalElement ABSTRACT
MAY ( dlmLength $ dlmMaxLength $ dlmMediaType $
      dlmWired )
)

( 1.3.6.1.4.1.412.100.3.1.2.41 NAME 'dlm1PhysicalLinkAuxClass'
  DESC 'The PhysicalLink class represents the cabling of
        PhysicalElements together. For example, serial or
        Ethernet cables, and infrared Links would be
        subclasses (if additional properties or associations
        are defined) or instances of PhysicalLink. In many
        cases, the numerous physical cables within a
        PhysicalPackage or Network will not be modeled.
        However, where these cables or Links are critical
        components, or are tagged assets of the company, these
        objects can be instantiated using this class or one of
        its descendent classes.'
  SUP dlm1PhysicalLink AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.42 NAME 'dlm1PhysicalLinkInstance'
  DESC 'The PhysicalLink class represents the cabling of
        PhysicalElements together. For example, serial or
        Ethernet cables, and infrared Links would be
        subclasses (if additional properties or associations
        are defined) or instances of PhysicalLink. In many
        cases, the numerous physical cables within a
        PhysicalPackage or Network will not be modeled.
        However, where these cables or Links are critical
        components, or are tagged assets of the company, these
        objects can be instantiated using this class or one of
        its descendent classes.'
  SUP dlm1PhysicalLink
)

( 1.3.6.1.4.1.412.100.3.3.2.14 NAME
'dlm1PhysicalLinkInstanceNameForm'
  OC dlm1PhysicalLinkInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-14> NAME 'dlm1PhysicalLinkInstanceStructureRule'
  Form dlm1PhysicalLinkInstanceNameForm
)
```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalLinkInstance.

```
( 1.3.6.1.4.1.412.100.3.1.2.42 NAME
'dlm1PhysicalLinkInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1PhysicalLinkInstance.'
  AUX ( dlm1ElementsLinkedAuxClass $ dlm1RealizesAuxClass
        $ dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
```

```

    dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
    dlm1ProvidesServiceToElementAuxClass $
    dlm1StatisticsAuxClass $
    dlm1ProductPhysicalElementsAuxClass $
    dlm1FRUPhysicalElementsAuxClass $
    dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
    $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
    dlm1SystemComponentAuxClass $
    dlm1MemberOfCollectionAuxClass $
    dlm1LinkHasConnectorAuxClass )
)

```

3.15 PhysicalConnector Classes

These classes represent any PhysicalElement that is used to connect to other Elements. Any object (for example, Slots and D-Shell connectors) that can be used to connect and transmit signals or power between two or more PhysicalElements is descendant of these classes.

```

( 1.3.6.1.4.1.412.100.3.2.1064 NAME 'dlmConnectorPinout'
DESC 'A free-form string describing the pin
configuration and signal usage of a PhysicalConnector.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1065 NAME 'dlmConnectorType'
DESC 'An array of integers defining the type of
PhysicalConnector. An array is specified to allow the
description of "combinations" of Connector
information. For example, one array entry could
specify RS-232 (value=25), another DB-25 (value=23)
and a third entry define the Connector as "Male"
(value=2). Values are 0="Unknown", 1="Other",
2="Male", 3="Female", 4="Shielded", 5="Unshielded",
6="SCSI (A) High-Density (50 pins)", 7="SCSI (A)
Low-Density (50 pins)", 8="SCSI (P) High-Density (68
pins)", 9="SCSI SCA-I (80 pins)", 10="SCSI SCA-II (80
pins)", 11="Fibre Channel (DB-9, 12=Copper)",
13="Fibre Channel (Optical Fibre)", 14="Fibre Channel
SCA-II (40 pins)", 15="Fibre Channel SCA-II (20
pins)", 16="Fibre Channel BNC", 17="ATA 3-1/2 Inch (40
pins)", 18="ATA 2-1/2 Inch (44 pins)", 19="ATA-2",
20="ATA-3", 21="ATA/66", 22="DB-9", 23="DB-15",
24="DB-25", 25="DB-36", 26="RS-232C", 27="RS-422",
28="RS-423", 29="RS-485", 30="RS-449", 31="V.35",
32="X.21", 33="IEEE-488", 34="AUI", 35="UPT Category
3", 36="UPT Category 4", 37="UPT Category 5",
38="BNC", 39="RJ11", 40="RJ45", 41="Fiber MIC",
42="Apple AUI", 43="Apple GeoPort", 44="PCI",
45="ISA", 46="EISA", 47="VESA", 48="PCMCIA",
49="PCMCIA Type I", 50="PCMCIA Type II", 51="PCMCIA
Type III", 52="ZV Port", 53="CardBus", 54="USB",
55="IEEE 1394", 56="HIPPI", 57="HSSDC (6 pins)",
58="GBIC", 59="DIN", 60="Mini-DIN", 61="Micro-DIN",
62="PS/2", 63="Infrared", 64="HP-HIL",
65="Access.bus", 66="NuBus", 67="Centronics",
)

```


DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
68="Mini-Centronics", 69="Mini-Centronics Type-14",
70="Mini-Centronics Type-20", 71="Mini-Centronics
Type-26", 72="Bus Mouse", 73="ADB", 74="AGP", 75="VME
Bus", 76="VME64", 77="Proprietary", 78="Proprietary
Processor Card Slot", 79="Proprietary Memory Card
Slot", 80="Proprietary I/O Riser Slot",
81="PCI-66MHZ", 82="AGP2X", 83="AGP4X", 84="PC-98",
85="PC-98-Hireso", 86="PC-H98", 87="PC-98Note",
88="PC-98Full", 89="SSA SCSI", 90="Circular", 91="On
Board IDE Connector", 92="On Board Floppy Connector",
93="9 Pin Dual Inline", 94="25 Pin Dual Inline",
95="50 Pin Dual Inline", 96="68 Pin Dual Inline",
97="On Board Sound Connector", 98="Mini-jack",
99="PCI-X", 100="Sbus IEEE 1396-1993 32 bit",
101="Sbus IEEE 1396-1993 64 bit", 102="MCA",
103="GIO", 104="XIO", 105="HIO", 106="NGIO",
107="PMC", 108="MTRJ", 109="VF-45", 110="Future I/O",
111="SC", 112="SG", 113="Electrical", 114="Optical",
115="Ribbon", 116="GLM", 117="1x9", 118="Mini SG",
119="LC", 120="HSSC", 121="VHDCI Shielded (68 pins)",
122="InfiniBand".'
```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

(1.3.6.1.4.1.412.100.3.2.1066 NAME 'dlmOtherTypeDescription'
DESC 'A string describing the Connector - used when the
ConnectorType property is set to 1 ("Other"). OtherType
Description should be set to NULL when ConnectorType is
any value other than 1.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
EQUALITY caseIgnoreMatch
)

(1.3.6.1.4.1.412.100.3.1.2.43 NAME 'dlm1PhysicalConnector'
DESC 'The PhysicalConnector class represents any
PhysicalElement that is used to connect to other
Elements. Any object that can be used to connect and
transmit signals or power between two or more
PhysicalElements is a descendant (or member) of this
class. For example, Slots and D-shell connectors are
types of PhysicalConnectors.'
SUP dlm1PhysicalElement ABSTRACT
MAY (dlmConnectorPinout \$ dlmConnectorType \$
dlmOtherTypeDescription)
)

(1.3.6.1.4.1.412.100.3.1.2.44 NAME 'dlm1PhysicalConnectorAuxClass'
DESC 'The PhysicalConnector class represents any
PhysicalElement that is used to connect to other
Elements. Any object that can be used to connect and
transmit signals or power between two or more
PhysicalElements is a descendant (or member) of this
class. For example, Slots and D-shell connectors are
types of PhysicalConnectors.'
SUP dlm1PhysicalConnector AUXILIARY
)

```

( 1.3.6.1.4.1.412.100.3.1.2.45 NAME 'dlm1PhysicalConnectorInstance'
  DESC 'The PhysicalConnector class represents any
        PhysicalElement that is used to connect to other
        Elements. Any object that can be used to connect and
        transmit signals or power between two or more
        PhysicalElements is a descendant (or member) of this
        class. For example, Slots and D-shell connectors are
        types of PhysicalConnectors.'
  SUP dlm1PhysicalConnector
)

( 1.3.6.1.4.1.412.100.3.3.2.15 NAME
'dlm1PhysicalConnectorInstanceNameForm'
  OC dlm1PhysicalConnectorInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-15> NAME 'dlm1PhysicalConnectorInstanceStructureRule'
  Form dlm1PhysicalConnectorInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1PhysicalConnectorInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.45 NAME
'dlm1PhysicalConnectorInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1PhysicalConnectorInstance.'
  AUX ( dlm1ConnectedToAuxClass $
        dlm1PackageInConnectorAuxClass $
        dlm1AdapterActiveConnectionAuxClass $
        dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1LinkHasConnectorAuxClass $
        dlm1ConnectorOnPackageAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass )
)

```

3.16 Slot Classes

These classes represent Connectors into which Packages are inserted.

```

( 1.3.6.1.4.1.412.100.3.2.1067 NAME 'dlmHeightAllowed'
  DESC 'Maximum height of an adapter Card that can be
        inserted into the Slot, in inches. The value is
        considered to be Inches.'
)

```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
SUP cimFloat32 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1068 NAME 'dlmLengthAllowed'
  DESC 'Maximum length of an adapter Card that can be
        inserted into the Slot, in inches. The value is
        considered to be Inches.'
  SUP cimFloat32 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1069 NAME 'dlmMaxDataWidth'
  DESC 'Maximum bus width of adapter Cards that can be
        inserted into this Slot, in bits. If the value is
        "unknown", enter 0. If the value is other than 8, 16,
        32, 64 or 128, enter 1. Value Mappings are "0", "1",
        "8", "16", "32", "64", "128" The value is considered
        to be Bits.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1070 NAME 'dlmNumber'
  DESC 'The Number property indicates the physical slot
        number, which can be used as an index into a system
        slot table, whether or not that slot is physically
        occupied.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1071 NAME 'dlmOpenSwitch'
  DESC 'A boolean indicating whether the switch state of
        the Slot is currently open (TRUE) or closed (FALSE).
        This switch state determines whether the contents of
        the Slot can be hot-plugged.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1072 NAME 'dlmPowered'
  DESC 'A boolean indicating whether the Slot is
        currently powered (TRUE) or not (FALSE).'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1073 NAME 'dlmPurposeDescription'
  DESC 'A free-form string describing that this Slot is
        physically unique and may hold special types of
        hardware. This property only has meaning when the
        corresponding boolean property, SpecialPurpose, is set
        to TRUE.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1074 NAME 'dlmSpecialPurpose'
  DESC 'Boolean indicating that this Slot is physically
        unique and may hold special types of hardware, e.g. a
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
        graphics processor slot. If set to TRUE, then the
        property, Special PurposeDescription (a string),
        should specify the nature of the uniqueness or purpose
        of the Slot.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1075 NAME 'dlmSupportsHotPlug'
  DESC 'Boolean indicating whether the Slot supports
        hot-plug of adapter Cards.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

( 1.3.6.1.4.1.412.100.3.2.1076 NAME 'dlmThermalRating'
  DESC 'Maximum thermal dissipation of the Slot in
        milliwatts. The value is considered to be MilliWatts.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1077 NAME 'dlmVccMixedVoltageSupport'
  DESC 'An array of enumerated integers indicating the
        Vcc voltage supported by this Slot. Values are
        0="Unknown", 1="Other", 2="3.3V", 3="5V".'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1078 NAME 'dlmVppMixedVoltageSupport'
  DESC 'An array of enumerated integers indicating the
        Vpp voltage supported by this Slot. Values are
        0="Unknown", 1="Other", 2="3.3V", 3="5V", 4="12V".'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.46 NAME 'dlm1Slot'
  DESC 'The Slot class represents Connectors into which
        Packages are inserted. For example, a PhysicalPackage
        that is a DiskDrive may be inserted into an SCA
        "Slot". As another example, a Card (subclass of
        PhysicalPackage) may be inserted into a 16-, 32-, or
        64-bit expansion "Slot" on a HostingBoard. PCI or
        PCMCIA Type III Slots are examples of the latter.'
```

```
SUP dlm1PhysicalConnector ABSTRACT
MAY ( dlmConnectorType $ dlmHeightAllowed $
      dlmLengthAllowed $ dlmMaxDataWidth $ dlmNumber $
      dlmOpenSwitch $ dlmPowered $ dlmPurposeDescription $
      dlmSpecialPurpose $ dlmSupportsHotPlug $
      dlmThermalRating $ dlmVccMixedVoltageSupport $
      dlmVppMixedVoltageSupport )
)

( 1.3.6.1.4.1.412.100.3.1.2.47 NAME 'dlm1SlotAuxClass'
  DESC 'The Slot class represents Connectors into which
        Packages are inserted. For example, a PhysicalPackage
        that is a DiskDrive may be inserted into an SCA
```

```

        "Slot". As another example, a Card (subclass of
        PhysicalPackage) may be inserted into a 16-, 32-, or
        64-bit expansion "Slot" on a HostingBoard. PCI or
        PCMCIA Type III Slots are examples of the latter.'
    SUP dlm1Slot AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.48 NAME 'dlm1SlotInstance'
  DESC 'The Slot class represents Connectors into which
  Packages are inserted. For example, a PhysicalPackage
  that is a DiskDrive may be inserted into an SCA
  "Slot". As another example, a Card (subclass of
  PhysicalPackage) may be inserted into a 16-, 32-, or
  64-bit expansion "Slot" on a HostingBoard. PCI or
  PCMCIA Type III Slots are examples of the latter.'
  SUP dlm1Slot
)

( 1.3.6.1.4.1.412.100.3.3.2.16 NAME 'dlm1SlotInstanceNameForm'
  OC dlm1SlotInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-16> NAME 'dlm1SlotInstanceStructureRule'
  Form dlm1SlotInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1SlotInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.48 NAME 'dlm1SlotInstanceContentRule'
  DESC 'Aux classes that can attach to dlm1SlotInstance.'
  AUX ( dlm1SlotInSlotAuxClass $
        dlm1AdjacentSlotsHelperAuxClass $
        dlm1PackageInSlotAuxClass $ dlm1CardInSlotAuxClass $
        dlm1ConnectedToAuxClass $
        dlm1PackageInConnectorAuxClass $
        dlm1AdapterActiveConnectionAuxClass $
        dlm1RealizesAuxClass $
        dlm1PhysicalElementLocationAuxClass $
        dlm1ElementCapacityAuxClass $
        dlm1ElementsLinkedAuxClass $
        dlm1ElementConfigurationAuxClass $
        dlm1ElementSettingAuxClass $
        dlm1DefaultSettingAuxClass $ dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $ dlm1LinkHasConnectorAuxClass $
        dlm1ConnectorOnPackageAuxClass $
        dlm1ProductPhysicalElementsAuxClass $
        dlm1FRUPhysicalElementsAuxClass $
        dlm1ParticipatesInSetAuxClass $ dlm1ContainerAuxClass $
        $ dlm1CollectedMSEsAuxClass $ dlm1ComponentAuxClass $
        dlm1SystemComponentAuxClass $
        dlm1MemberOfCollectionAuxClass )
)

```

3.17 MediaPhysicalStatInfo Classes

These classes hold statistics related to reading physical labels and picks/puts at a specific StorageMediaLocatino or for a specific piece of PhysicalMedia.

```
( 1.3.6.1.4.1.412.100.3.2.1079 NAME 'dlnPickFailures'  
  DESC 'The number of failed picks.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1080 NAME 'dlnPickRetries'  
  DESC 'The number of retried picks.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1081 NAME 'dlnPickSuccesses'  
  DESC 'The number of successful picks.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1082 NAME 'dlnPutFailures'  
  DESC 'The number of failed puts.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1083 NAME 'dlnPutRetries'  
  DESC 'The number of retried puts.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1084 NAME 'dlnPutSuccesses'  
  DESC 'The number of successful puts.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1085 NAME 'dlnScanFailures'  
  DESC 'The number of failed physical label scans.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1086 NAME 'dlnScanRetries'  
  DESC 'The number of retried physical label scans.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)  
  
( 1.3.6.1.4.1.412.100.3.2.1087 NAME 'dlnScanSuccesses'  
  DESC 'The number of successful physical label scans.'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE  
  EQUALITY integerMatch  
)
```

DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
( 1.3.6.1.4.1.412.100.3.1.2.49 NAME 'dlm1MediaPhysicalStatInfo'  
  DESC 'Statistics related to reading physical labels and  
        picks/puts at a specific StorageMediaLocation, or for a  
        specific piece of PhysicalMedia. Although the same  
        class is used to represent this data, at the instance  
        level the object holds information for the Location  
        (independent of the Media), OR for the Media  
        (independent of its StorageMediaLocation).'  SUP dlm1PhysicalStatisticalInformation ABSTRACT  
  MAY ( $ dlmPickFailures $ dlmPickRetries $  
        dlmPickSuccesses $ dlmPutFailures $ dlmPutRetries $  
        dlmPutSuccesses $ dlmScanFailures $ dlmScanRetries $  
        dlmScanSuccesses )  
)  
  
( 1.3.6.1.4.1.412.100.3.1.2.50 NAME  
'dlm1MediaPhysicalStatInfoAuxClass'  
  DESC 'Statistics related to reading physical labels and  
        picks/puts at a specific StorageMediaLocation, or for a  
        specific piece of PhysicalMedia. Although the same  
        class is used to represent this data, at the instance  
        level the object holds information for the Location  
        (independent of the Media), OR for the Media  
        (independent of its StorageMediaLocation).'  SUP dlm1MediaPhysicalStatInfo AUXILIARY  
)  
  
( 1.3.6.1.4.1.412.100.3.1.2.51 NAME  
'dlm1MediaPhysicalStatInfoInstance'  
  DESC 'Statistics related to reading physical labels and  
        picks/puts at a specific StorageMediaLocation, or for a  
        specific piece of PhysicalMedia. Although the same  
        class is used to represent this data, at the instance  
        level the object holds information for the Location  
        (independent of the Media), OR for the Media  
        (independent of its StorageMediaLocation).'  SUP dlm1MediaPhysicalStatInfo  
)  
  
( 1.3.6.1.4.1.412.100.3.3.2.17 NAME  
'dlm1MediaPhysicalStatInfoInstanceNameForm'  
  OC dlm1MediaPhysicalStatInfoInstance  
  MUST ( orderedCimKeys )  
)  
  
( <physical-sr-17> NAME  
'dlm1MediaPhysicalStatInfoInstanceStructureRule'  
  Form dlm1MediaPhysicalStatInfoInstanceNameForm  
)
```

The following content rule shows what auxiliary classes may be attached to dlm1MediaPhysicalStatInfoInstance.

```
( 1.3.6.1.4.1.412.100.3.1.2.51 NAME  
'dlm1MediaPhysicalStatInfoInstanceContentRule'  
  DESC 'Aux classes that can attach to
```

```

        dlm1MediaPhysicalStatInfoInstance.'
AUX ( dlm1StatisticsAuxClass $
      dlm1RelatedStatisticsAuxClass $ dlm1DependencyAuxClass
      $ dlm1ProvidesServiceToElementAuxClass $
      dlm1MemberOfCollectionAuxClass )
)

```

3.18 Location Classes

These classes specify the position and address of a PhysicalElement.

```

( 1.3.6.1.4.1.412.100.3.2.1088 NAME 'dlmAddress'
  DESC 'Address is a free-form string indicating a
        street, building or other type of address for the
        PhysicalElement's Location.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024} SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.2.1089 NAME 'dlmPhysicalPosition'
  DESC 'Position is a free-form string indicating the
        placement of a PhysicalElement. It can specify slot
        information on a HostingBoard, mounting site in a
        Cabinet, or latitude and longitude information, for
        example, from a GPS. It is part of the key of the
        Location object.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} SINGLE-VALUE
  EQUALITY caseIgnoreMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.52 NAME 'dlm1Location'
  DESC 'The Location class specifies the position and
        address of a PhysicalElement.'
  SUP dlm1ManagedElement ABSTRACT
  MAY ( dlmAddress $ dlmName $ dlmPhysicalPosition )
)

( 1.3.6.1.4.1.412.100.3.1.2.53 NAME 'dlm1LocationAuxClass'
  DESC 'The Location class specifies the position and
        address of a PhysicalElement.'
  SUP dlm1Location AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.54 NAME 'dlm1LocationInstance'
  DESC 'The Location class specifies the position and
        address of a PhysicalElement.'
  SUP dlm1Location
)

( 1.3.6.1.4.1.412.100.3.3.2.18 NAME 'dlm1LocationInstanceNameForm'
  OC dlm1LocationInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-18> NAME 'dlm1LocationInstanceStructureRule'
  Form dlm1LocationInstanceNameForm
)

```


The following content rule shows what auxiliary classes may be attached to dlm1LocationInstance.

```
( 1.3.6.1.4.1.412.100.3.1.2.54 NAME 'dlm1LocationInstanceContentRule'
DESC 'Aux classes that can attach to
    dlm1LocationInstance.'
AUX ( dlm1PhysicalElementLocationAuxClass $
      dlm1DependencyAuxClass $
      dlm1ProvidesServiceToElementAuxClass $
      dlm1StatisticsAuxClass $
      dlm1MemberOfCollectionAuxClass )
)
```

3.19 PhysicalCapacity

This abstract class describes a PhysicalElement's minimum/maximum requirements and ability to support different types of hardware. For example, minimum and maximum memory requirements can be modeled as a subclass of PhysicalCapacity

```
( 1.3.6.1.4.1.412.100.3.1.2.55 NAME 'dlm1PhysicalCapacity'
DESC 'PhysicalCapacity is an abstract class describing
    a PhysicalElement"s minimum/maximum requirements and
    ability to support different types of hardware. For
    example, minimum and maximum memory requirements can
    be modeled as a subclass of PhysicalCapacity.'
SUP dlm1ManagedElement ABSTRACT
MAY ( dlmName )
)
```

3.20 MemoryCapacity Classes

These classes describe the type of Memory that can be installed on a PhysicalElement and its minimum/maximum configurations. The PhysicalMemory class models information about what memory is currently installed.

```
( 1.3.6.1.4.1.412.100.3.2.1091 NAME 'dlmMaximumMemoryCapacity'
DESC 'Maximum amount of memory, in Kbytes, that can be
    supported by the associated PhysicalElement. The
    value is considered to be KiloBytes.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.2.1092 NAME 'dlmMinimumMemoryCapacity'
DESC 'Minimum amount of memory, in Kbytes, that is
    needed for the associated PhysicalElement to operate
    correctly. The value is considered to be KiloBytes.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

( 1.3.6.1.4.1.412.100.3.1.2.56 NAME 'dlm1MemoryCapacity'
DESC 'MemoryCapacity describes the type of Memory that
    can be installed on a PhysicalElement and its
    minimum/maximum configurations. Information on what
```

```

        memory is currently "installed", versus an Element's
        min/max requirements, is located in instances of the
        PhysicalMemory class.'
    SUP dlm1PhysicalCapacity ABSTRACT
    MAY ( dlmMaximumMemoryCapacity $ dlmMemoryType $
          dlmMinimumMemoryCapacity $ dlmName )
)

( 1.3.6.1.4.1.412.100.3.1.2.57 NAME 'dlm1MemoryCapacityAuxClass'
  DESC 'MemoryCapacity describes the type of Memory that
        can be installed on a PhysicalElement and its
        minimum/maximum configurations. Information on what
        memory is currently "installed", versus an Element's
        min/max requirements, is located in instances of the
        PhysicalMemory class.'
  SUP dlm1MemoryCapacity AUXILIARY
)

( 1.3.6.1.4.1.412.100.3.1.2.58 NAME 'dlm1MemoryCapacityInstance'
  DESC 'MemoryCapacity describes the type of Memory that
        can be installed on a PhysicalElement and its
        minimum/maximum configurations. Information on what
        memory is currently "installed", versus an Element's
        min/max requirements, is located in instances of the
        PhysicalMemory class.'
  SUP dlm1MemoryCapacity
)

( 1.3.6.1.4.1.412.100.3.3.2.19 NAME
'dlm1MemoryCapacityInstanceNameForm'
  OC dlm1MemoryCapacityInstance
  MUST ( orderedCimKeys )
)

( <physical-sr-19> NAME 'dlm1MemoryCapacityInstanceStructureRule'
  Form dlm1MemoryCapacityInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1MemoryCapacityInstance.

```

( 1.3.6.1.4.1.412.100.3.1.2.58 NAME
'dlm1MemoryCapacityInstanceContentRule'
  DESC 'Aux classes that can attach to
        dlm1MemoryCapacityInstance.'
  AUX ( dlm1ElementCapacityAuxClass $
        dlm1DependencyAuxClass $
        dlm1ProvidesServiceToElementAuxClass $
        dlm1StatisticsAuxClass $
        dlm1MemberOfCollectionAuxClass )
)

```

3.21 ConfigurationCapacity Classes

These classes provide information on minimum and maximum of power supplies, fans, disk drives, etc. that can be connected to or placed on/into

- a PhysicalElement (as well as the number that must be connected/added/removed at a time).**
- ```
(1.3.6.1.4.1.412.100.3.2.1093 NAME 'dlmIncrement'
 DESC 'Increment in which Elements must be added or
 removed.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
 EQUALITY integerMatch
)
```
- ```
( 1.3.6.1.4.1.412.100.3.2.1094 NAME 'dlmMaximumCapacity'
  DESC 'Maximum number of Elements of type, ObjectType,
        that may be installed.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)
```
- ```
(1.3.6.1.4.1.412.100.3.2.1095 NAME 'dlmMinimumCapacity'
 DESC 'Minimum number of Elements of type, ObjectType,
 that must be installed.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
 EQUALITY integerMatch
)
```
- ```
( 1.3.6.1.4.1.412.100.3.2.1096 NAME 'dlmObjectType'
  DESC 'The type of object (power supply, fan, disk
        drive, ...) whose capacities are indicated. This
        information is part of the class" key. Values are
        0="Other", 1="Processors", 2="Power Supplies",
        3="Fans", 4="Batteries", 5="I/O Slots", 6="Memory
        Slots", 7="MediaAccessDevices (Drives)",
        8="StorageMediaLocation Slots",
        9="StorageMediaLocation Magazines",
        10="StorageMediaLocation Panels",
        11="StorageMediaLocation InterLibrary Ports",
        12="StorageMediaLocation Limited Access Ports",
        13="Doors", 14="MediaTransferDevice Pickers",
        15="MediaTransferDevice Changers", 16="LabelReaders",
        17="Contained Chassis", 18="Connected Chassis",
        19="Connected Frames", 20="Front Side I/O Slots",
        21="Back Side I/O Slots", 22="Cache Memory", 23="NVS
        Memory", 24="Volatile Memory".'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
  EQUALITY integerMatch
)
```
- ```
(1.3.6.1.4.1.412.100.3.1.2.59 NAME 'dlm1ConfigurationCapacity'
 DESC 'ConfigurationCapacity provides information on the
 minimum and maximum numbers of power supplies, fans,
 disk drives, etc. that can be connected to or placed
 on/into a PhysicalElement (and the number that must be
 connected/added/removed at a time). The PhysicalElement
 whose configuration is described is identified using
 the ElementCapacity association, inherited from
 PhysicalCapacity. The object whose capacities are
 indicated (ie, the power supply or fan) is identified
 in the ObjectType property of this class. Since the
```

## DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
 same min/max configurations can apply to multiple
 instances, this class is not defined as "weak".
 Examples of the use of the ConfigurationCapacity class
 are to describe that a "control unit" Chassis may be
 connected to (at most) 4 other I/O chassis, or to
 describe what a StorageLibrary"s cabinet may contain.
 Continuing the latter example, a particular
 StorageLibrary"s cabinet might hold a minimum of 3 and
 a maximum of 9 TapeDrives, and a minimum of 88 and a
 maximum of 264 StorageMediaLocations (\\"Slots\\'
SUP dlm1PhysicalCapacity ABSTRACT
MAY (dlmIncrement $ dlmMaximumCapacity $
 dlmMinimumCapacity $ dlmName $ dlmObjectType $
 dlmOtherTypeDescription)
)

(1.3.6.1.4.1.412.100.3.1.2.60 NAME
'dlm1ConfigurationCapacityAuxClass'
DESC 'ConfigurationCapacity provides information on the
minimum and maximum numbers of power supplies, fans,
disk drives, etc. that can be connected to or placed
on/into a PhysicalElement (and the number that must be
connected/added/removed at a time). The PhysicalElement
whose configuration is described is identified using
the ElementCapacity association, inherited from
PhysicalCapacity. The object whose capacities are
indicated (ie, the power supply or fan) is identified
in the ObjectType property of this class. Since the
same min/max configurations can apply to multiple
instances, this class is not defined as "weak".
Examples of the use of the ConfigurationCapacity class
are to describe that a "control unit" Chassis may be
connected to (at most) 4 other I/O chassis, or to
describe what a StorageLibrary"s cabinet may contain.
Continuing the latter example, a particular
StorageLibrary"s cabinet might hold a minimum of 3 and
a maximum of 9 TapeDrives, and a minimum of 88 and a
maximum of 264 StorageMediaLocations (\\"Slots\\'
SUP dlm1ConfigurationCapacity AUXILIARY
)

(1.3.6.1.4.1.412.100.3.1.2.61 NAME
'dlm1ConfigurationCapacityInstance'
DESC 'ConfigurationCapacity provides information on the
minimum and maximum numbers of power supplies, fans,
disk drives, etc. that can be connected to or placed
on/into a PhysicalElement (and the number that must be
connected/added/removed at a time). The PhysicalElement
whose configuration is described is identified using
the ElementCapacity association, inherited from
PhysicalCapacity. The object whose capacities are
indicated (ie, the power supply or fan) is identified
in the ObjectType property of this class. Since the
same min/max configurations can apply to multiple
instances, this class is not defined as "weak".
Examples of the use of the ConfigurationCapacity class
are to describe that a "control unit" Chassis may be
```

```

 connected to (at most) 4 other I/O chassis, or to
 describe what a StorageLibrary"s cabinet may contain.
 Continuing the latter example, a particular
 StorageLibrary"s cabinet might hold a minimum of 3 and
 a maximum of 9 TapeDrives, and a minimum of 88 and a
 maximum of 264 StorageMediaLocations (\\"Slots\\"
SUP dlm1ConfigurationCapacity
)

(1.3.6.1.4.1.412.100.3.3.2.20 NAME
'dlm1ConfigurationCapacityInstanceNameForm'
 OC dlm1ConfigurationCapacityInstance
 MUST (orderedCimKeys)
)

(<physical-sr-20> NAME
'dlm1ConfigurationCapacityInstanceStructureRule'
 Form dlm1ConfigurationCapacityInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to `dlm1ConfigurationCapacityInstance`.

```

(1.3.6.1.4.1.412.100.3.1.2.61 NAME
'dlm1ConfigurationCapacityInstanceContentRule'
 DESC 'Aux classes that can attach to
 dlm1ConfigurationCapacityInstance.'
 AUX (dlm1ElementCapacityAuxClass $
 dlm1DependencyAuxClass $
 dlm1ProvidesServiceToElementAuxClass $
 dlm1StatisticsAuxClass $
 dlm1MemberOfCollectionAuxClass)
)

```

### 3.22 ReplacementSet Classes

These classes aggregate `PhysicalElements` that must be replaced or FRUed together. Examples are replacing the component memory chips of a memory card when the card is replaced or when just replacing the component memory chips as a set.

```

(1.3.6.1.4.1.412.100.3.1.2.62 NAME 'dlm1ReplacementSet'
 DESC 'The ReplacementSet class aggregates
 PhysicalElements that must be "replaced" or "FRUed"
 together. For example, when replacing a memory card,
 the component memory chips could be removed and
 replaced as well. Or, a set of memory chips may be
 specified to be replaced or upgraded together using
 this association.'
 SUP dlm1ManagedElement ABSTRACT
 MAY (dlmName)
)

(1.3.6.1.4.1.412.100.3.1.2.63 NAME 'dlm1ReplacementSetAuxClass'
 DESC 'The ReplacementSet class aggregates
 PhysicalElements that must be "replaced" or "FRUed"
 together. For example, when replacing a memory card,

```

```

 the component memory chips could be removed and
 replaced as well. Or, a set of memory chips may be
 specified to be replaced or upgraded together using
 this association.'
 SUP dlm1ReplacementSet AUXILIARY
)

(1.3.6.1.4.1.412.100.3.1.2.64 NAME 'dlm1ReplacementSetInstance'
 DESC 'The ReplacementSet class aggregates
 PhysicalElements that must be "replaced" or "FRUed"
 together. For example, when replacing a memory card,
 the component memory chips could be removed and
 replaced as well. Or, a set of memory chips may be
 specified to be replaced or upgraded together using
 this association.'
 SUP dlm1ReplacementSet
)

(1.3.6.1.4.1.412.100.3.3.2.21 NAME
'dlm1ReplacementSetInstanceNameForm'
 OC dlm1ReplacementSetInstance
 MUST (orderedCimKeys)
)

(<physical-sr-21> NAME 'dlm1ReplacementSetInstanceStructureRule'
 Form dlm1ReplacementSetInstanceNameForm
)

```

The following content rule shows what auxiliary classes may be attached to dlm1ReplacementSetInstance.

```

(1.3.6.1.4.1.412.100.3.1.2.64 NAME
'dlm1ReplacementSetInstanceContentRule'
 DESC 'Aux classes that can attach to
 dlm1ReplacementSetInstance.'
 AUX (dlm1DependencyAuxClass $
 dlm1ProvidesServiceToElementAuxClass $
 dlm1StatisticsAuxClass $
 dlm1MemberOfCollectionAuxClass $
 dlm1ParticipatesInSetAuxClass)
)

```

### 3.23 RealizesExtent Classes

As PhysicalComponents can realize StorageExtents, these classes make this realization explicit. In addition, the StartingAddress of the StorageExtent on the Component is specified here.

```

(1.3.6.1.4.1.412.100.3.2.1097 NAME 'dlmStartingAddress'
 DESC 'The starting address on the PhysicalComponent
 where the StorageExtent begins. Ending address of the
 StorageExtent is determined using the NumberOfBlocks
 and Block Size properties of the StorageExtent object.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
 EQUALITY integerMatch
)

(1.3.6.1.4.1.412.100.3.1.2.65 NAME 'dlm1RealizesExtent'

```

```

DESC 'StorageExtents can be realized by
PhysicalComponents. For example, disks or tapes are
realized by PhysicalMedia. Memory is realized by
PhysicalMemory. This relationship of Extents to
PhysicalComponents is made explicit by the
RealizesExtent association. In addition, the
StartingAddress of the StorageExtent on the Component
is specified here.'
SUP dlm1Realizes ABSTRACT
MAY (dlmStartingAddress)
)

(1.3.6.1.4.1.412.100.3.2.1098 NAME 'dlmRealizesExtentAntecedentRef'
DESC 'The PhysicalComponent on which the Extent is
realized. The value of this attribute points to an
entry of class dlmPhysicalComponent.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1099 NAME 'dlmRealizesExtentDependentRef'
DESC 'The StorageExtent that is located on the
Component. Values of this attribute point to
entries of class dlmStorageExtent.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.66 NAME 'dlm1RealizesExtentAuxClass'
DESC 'StorageExtents can be realized by
PhysicalComponents. For example, disks or tapes are
realized by PhysicalMedia. Memory is realized by
PhysicalMemory. This relationship of Extents to
PhysicalComponents is made explicit by the
RealizesExtent association. In addition, the
StartingAddress of the StorageExtent on the Component
is specified here.'
SUP dlm1RealizesExtent AUXILIARY
MAY (dlmRealizesExtentAntecedentRef $
dlmRealizesExtentDependentRef)
)

```

### 3.24 RealizesPExtent Classes

The relationship between PhysicalExtents and PhysicalMedia is made explicit with these classes.

```

(1.3.6.1.4.1.412.100.3.1.2.67 NAME 'dlm1RealizesPExtent'
DESC 'PhysicalExtents are realized on a PhysicalMedia.
This relationship is made explicit by the
RealizesPExtent association. In addition, the
StartingAddress of the PhysicalExtent on the
PhysicalMedia is specified here.'
SUP dlm1RealizesExtent ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1100 NAME 'dlmRealizesPExtentAntecedentRef'

```

```

DESC 'The PhysicalMedia on which the Extent is
 realized. The value of this attribute points to an
 entry of class dlmPhysicalMedia.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1101 NAME 'dlmRealizesPEntentDependentRef'
DESC 'The PhysicalExtent that is located on the Media.
 Values of this attribute point to entries of class
 dlmPhysicalExtent.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.68 NAME 'dlm1RealizesPEntentAuxClass'
DESC 'PhysicalExtents are realized on a PhysicalMedia.
 This relationship is made explicit by the
 RealizesPEntent association. In addition, the
 StartingAddress of the PhysicalExtent on the
 PhysicalMedia is specified here.'
SUP dlm1RealizesPEntent AUXILIARY
MAY (dlmRealizesPEntentAntecedentRef $
 dlmRealizesPEntentDependentRef)
)

```

### 3.25 RealizesDiskPartition Classes

These classes provide an alternative to the DiskPartitionBasedOnVolume association for realizing DiskPartitions on PhysicalMedia directly, e.g. via the fdisk or pdisk commands.

```

(1.3.6.1.4.1.412.100.3.1.2.69 NAME 'dlm1RealizesDiskPartition'
DESC 'DiskPartitions can be directly realized on a
 PhysicalMedia. This is used to model the creation of
 Partitions on a raw SCSI or IDE drive, using the fdisk
 (DOS and Unix) or pdisk (Unix) command. The
 StartingAddress of the DiskPartition on the
 PhysicalMedia is also specified as a property of this
 relationship. An alternative is that Partitions can be
 BasedOn StorageVolumes, such as a VolumeSet or a Volume
 exposed by a hardware RAID cabinet. The latter
 relationship is modeled using the
 DiskPartitionBasedOnVolume association.'
SUP dlm1RealizesExtent ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1102 NAME
'dlmRealizesDiskPartitionAntecedentRef'
DESC 'The PhysicalMedia on which the Extent is
 realized. The value of this attribute points to an
 entry of class dlmPhysicalMedia.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1103 NAME

```



```
'dlmRealizesDiskPartitionDependentRef'
 DESC 'The DiskPartition that is located on the Media.
 Values of this attribute point to entries of class
 dlmDiskPartition.'
```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.12  
EQUALITY distinguishedNameMatch  
)

( 1.3.6.1.4.1.412.100.3.1.2.70 NAME  
'dlm1RealizesDiskPartitionAuxClass'  
DESC 'DiskPartitions can be directly realized on a  
PhysicalMedia. This is used to model the creation of  
Partitions on a raw SCSI or IDE drive, using the fdisk  
(DOS and Unix) or pdisk (Unix) command. The  
StartingAddress of the DiskPartition on the  
PhysicalMedia is also specified as a property of this  
relationship. An alternative is that Partitions can be  
BasedOn StorageVolumes, such as a VolumeSet or a Volume  
exposed by a hardware RAID cabinet. The latter  
relationship is modeled using the  
DiskPartitionBasedOnVolume association.'  
SUP dlm1RealizesDiskPartition AUXILIARY  
MAY ( dlmRealizesDiskPartitionAntecedentRef \$  
 dlmRealizesDiskPartitionDependentRef )  
)

### 3.26 RealizesAggregatePEExtent

These classes delineate the realization of AggregatePEExtents on a PhysicalMedia.

```
(1.3.6.1.4.1.412.100.3.1.2.71 NAME 'dlm1RealizesAggregatePEExtent'
DESC 'AggregatePEExtents are realized on a
PhysicalMedia. This relationship is made explicit by
the RealizesAggregatePEExtent association.'
SUP dlm1RealizesExtent ABSTRACT
)
```

( 1.3.6.1.4.1.412.100.3.2.1104 NAME  
'dlmRealizesAggregatePEExtentAntecedentRef'  
DESC 'The PhysicalMedia on which the Extent is  
realized. The value of this attribute points to  
an entry of class dlmPhysicalMedia.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE  
EQUALITY distinguishedNameMatch  
)

( 1.3.6.1.4.1.412.100.3.2.1105 NAME  
'dlmRealizesAggregatePEExtentDependentRef'  
DESC 'The AggregatePEExtent that is located on the  
Media. Values of this attribute point to entries  
of class dlmAggregatePEExtent.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12  
EQUALITY distinguishedNameMatch  
)

( 1.3.6.1.4.1.412.100.3.1.2.72 NAME

```
'dlm1RealizesAggregatePEExtentAuxClass'
 DESC 'AggregatePEExtents are realized on a
 PhysicalMedia. This relationship is made explicit by
 the RealizesAggregatePEExtent association.'
 SUP dlm1RealizesAggregatePEExtent AUXILIARY
 MAY (dlmRealizesAggregatePEExtentAntecedentRef $
 dlmRealizesAggregatePEExtentDependentRef)
)
```

### 3.27 RealizesTapePartition Classes

These classes model the realization of TapePartitions on a PhysicalTape.

```
(1.3.6.1.4.1.412.100.3.1.2.73 NAME 'dlm1RealizesTapePartition'
 DESC 'TapePartitions are realized on PhysicalTape. This
 relationship is made explicit by the
 RealizesTapePartition association.'
 SUP dlm1RealizesExtent ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1106 NAME
'dlmRealizesTapePartitionAntecedentRef'
 DESC 'The PhysicalTape on which the Partition is
 realized. The value of this attribute points to an
 entry of class dlmPhysicalTape.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1107 NAME
'dlmRealizesTapePartitionDependentRef'
 DESC 'The TapePartition that is located on the Media.
 Values of this attribute point to entries of class
 dlmTapePartition.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.74 NAME
'dlm1RealizesTapePartitionAuxClass'
 DESC 'TapePartitions are realized on PhysicalTape. This
 relationship is made explicit by the
 RealizesTapePartition association.'
 SUP dlm1RealizesTapePartition AUXILIARY
 MAY (dlmRealizesTapePartitionAntecedentRef $
 dlmRealizesTapePartitionDependentRef)
)
```

### 3.28 RealizedOnSide Classes

Since removable PhysicalMedia can be dual-sided, these classes cover the scenario where StorageExtents are realized on a single side of the Media.

```
(1.3.6.1.4.1.412.100.3.2.1108 NAME 'dlmSide'
 DESC 'An enumeration expressing on which "Side" the
 Extent is realized. Since sides can be named by
```

```

 various schemes (0/1 or A/B), both schemes are
 expressed in the Values array of this property.
 Values are 0="Unknown", 1="Side 0", 2="Side 1",
 3="Side A", 4="Side B".'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

(1.3.6.1.4.1.412.100.3.1.2.75 NAME 'dlm1RealizedOnSide'
DESC 'Since removable PhysicalMedia can be dual-sided,
there is the possibility for StorageExtents to be
realized on a single side of the Media. This
association is a specialization of the RealizesExtent
relationship, adding a Side property to express the
Media"s orientation details.'
SUP dlm1RealizesExtent ABSTRACT
MAY (dlmSide)
)

(1.3.6.1.4.1.412.100.3.2.1109 NAME 'dlmRealizedOnSideAntecedentRef'
DESC 'The PhysicalMedia on which the Extent is
realized. Values of this attribute point to
entries of class dlmPhysicalMedia.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1110 NAME 'dlmRealizedOnSideDependentRef'
DESC 'The StorageExtent that is located on the
Component. Values of this attribute point to
entries of class dlmStorageExtent.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.76 NAME 'dlm1RealizedOnSideAuxClass'
DESC 'Since removable PhysicalMedia can be dual-sided,
there is the possibility for StorageExtents to be
realized on a single side of the Media. This
association is a specialization of the RealizesExtent
relationship, adding a Side property to express the
Media"s orientation details.'
SUP dlm1RealizedOnSide AUXILIARY
MAY (dlmRealizedOnSideAntecedentRef $
dlmRealizedOnSideDependentRef)
)

```

### 3.29 Docked Classes

These classes model the relationship between a laptop Chassis and a docking station Chassis.

```

(1.3.6.1.4.1.412.100.3.1.2.77 NAME 'dlm1Docked'
DESC 'A laptop, a type of Chassis, may be docked in
another type of Chassis, a Docking Station. This is
the relationship represented by the Docked
association. Because this is such a typical

```

```

 relationship, it is explicitly described.'
 SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1111 NAME 'dlmDockedAntecedentRef'
 DESC 'The Docking Station. The value of this attribute
 points to an entry of class dlmChassis.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1112 NAME 'dlmDockedDependentRef'
 DESC 'The Laptop that is "Docked". The value of this
 attribute points to an entry of class dlmChassis.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.78 NAME 'dlm1DockedAuxClass'
 DESC 'A laptop, a type of Chassis, may be docked in
 another type of Chassis, a Docking Station. This is
 the relationship represented by the Docked
 association. Because this is such a typical
 relationship, it is explicitly described.'
 SUP dlm1Docked AUXILIARY
 MAY (dlmDockedAntecedentRef $ dlmDockedDependentRef)
)

```

### 3.30 DeviceServicesLocation Classes

These classes indicate that the TransferDevice handles Media stored in the referenced Location. As an example LibraryPort "A" may only service Media from Slots 1-10 which LibraryPort "B" covers Slots 11-33.

```

(1.3.6.1.4.1.412.100.3.2.1113 NAME 'dlmInaccessible'
 DESC 'Boolean indicating that the referenced
 StorageMediaLocation is not currently accessible to
 the MediaTransferDevice. For example, the Location
 could be the realization of an InterLibraryPort that
 is currently servicing another Library. The
 DeviceServicesLocation association describes that the
 TransferDevice COULD service the MediaLocation. This
 boolean indicates that that this is temporarily not
 possible.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)

(1.3.6.1.4.1.412.100.3.1.2.79 NAME 'dlm1DeviceServicesLocation'
 DESC 'Within an automated StorageLibrary, Media should
 be accessible to the various robotics and
 MediaTransferDevices (Pickers, Changers,
 InterLibraryPorts, etc.). The Library may be serviced
 by different TransferDevices, each responsible for a
 subset of the Library"s StorageMediaLocations. The
 Device ServicesLocation association indicates that the
 Transfer Device handles Media stored in the referenced
 Location. For example, LibraryPort "A" may only

```

## DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
 service Media from Slots 1-10, while LibraryPort "B"
 covers Slots 11-33. This detail is conveyed by this
 association.'
```

```
 SUP dlm1Dependency ABSTRACT
 MAY (dlmInaccessible)
)

(1.3.6.1.4.1.412.100.3.2.1114 NAME
'dlmDeviceServicesLocationAntecedentRef'
 DESC 'The MediaTransferDevice that handles Media from
 the StorageMediaLocation. The value of this attribute
 points to an entry of class dlmMediaTransferDevice.'
```

```
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1115 NAME
'dlmDeviceServicesLocationDependentRef'
 DESC 'The MediaLocation that is serviced. The value of
 this attribute points to an entry of class
 dlmStorageMediaLocation.'
```

```
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.80 NAME
'dlm1DeviceServicesLocationInstance'
 DESC 'Within an automated StorageLibrary, Media should
 be accessible to the various robotics and
 MediaTransferDevices (Pickers, Changers,
 InterLibraryPorts, etc.). The Library may be serviced
 by different TransferDevices, each responsible for a
 subset of the Library"s StorageMediaLocations. The
 Device ServicesLocation association indicates that the
 Transfer Device handles Media stored in the referenced
 Location. For example, LibraryPort "A" may only
 service Media from Slots 1-10, while LibraryPort "B"
 covers Slots 11-33. This detail is conveyed by this
 association.'
```

```
 SUP dlm1DeviceServicesLocation
 MAY (dlmDeviceServicesLocationAntecedentRef $
 dlmDeviceServicesLocationDependentRef)
)

(1.3.6.1.4.1.412.100.3.3.2.22 NAME
'dlm1DeviceServicesLocationInstanceNameForm'
 OC dlm1DeviceServicesLocationInstance
 MUST (orderedCimKeys)
)

(<physical-sr-22> NAME
'dlm1DeviceServicesLocationInstanceStructureRule'
 Form dlm1DeviceServicesLocationInstanceNameForm
)

(1.3.6.1.4.1.412.100.3.2.1116 NAME
'dlmDeviceServicesLocationHelperRef'
```

```

DESC 'Pointer to DeviceServicesLocationInstance.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.81 NAME
'dlm1DeviceServicesLocationHelperAuxClass'
DESC 'Helper class for finding DeviceServicesLocation.'
SUP top AUXILIARY
MAY (dlmDeviceServicesLocationHelperRef)
)

```

### 3.31 MemoryWithMedia Classes

These classes indicate that Memory is associated with a Physical Media and its cartridge.

```

(1.3.6.1.4.1.412.100.3.1.2.82 NAME 'dlm1MemoryWithMedia'
DESC 'MemoryWithMedia indicates that Memory is
associated with a PhysicalMedia and its cartridge. The
Memory provides media identification and also stores
user-specific data. '
SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1117 NAME 'dlmMemoryWithMediaAntecedentRef'
DESC 'The Memory associated with PhysicalMedia. .
Values of this attribute points to entries of class
dlmPhysicalMemory.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1118 NAME 'dlmMemoryWithMediaDependentRef'
DESC 'The PhysicalMedia. Values of this attribute
point to entries of class dlmPhysicalMedia.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.83 NAME 'dlm1MemoryWithMediaAuxClass'
DESC 'MemoryWithMedia indicates that Memory is
associated with a PhysicalMedia and its cartridge. The
Memory provides media identification and also stores
user-specific data. '
SUP dlm1MemoryWithMedia AUXILIARY
MAY (dlmMemoryWithMediaAntecedentRef $
dlmMemoryWithMediaDependentRef)
)

```

### 3.32 PhysicalMediaInLocation Classes

These classes identify what Media is present in a StorageLocation. If this association does not exist, then the Location may be assumed to be empty.

```

(1.3.6.1.4.1.412.100.3.2.1119 NAME 'dlmOrientation'

```

## DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
DESC 'An enumeration expressing which "Side" of the
Media is facing up. Since sides can be named by
various schemes (0/1 or A/B), both schemes are
expressed in the Values array of this property. Also,
the value 5 ("Not Applicable" describes that
orientation is not pertinent for this Media. Values
are 0="Unknown", 1="Side 0", 2="Side 1", 3="Side A",
4="Side B", 5="Not Applicable".'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
EQUALITY integerMatch
)

(1.3.6.1.4.1.412.100.3.2.1120 NAME 'dlmTimeOfLastPlacement'
DESC 'The date and time that the Media was last placed
into the referenced Location.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 SINGLE-VALUE
EQUALITY generalizedTimeMatch
)

(1.3.6.1.4.1.412.100.3.1.2.84 NAME 'dlm1PhysicalMediaInLocation'
DESC 'Within a StorageLibrary, all Media should be
accounted for, and be present in some Storage
Location. This relationship is made explicit by the
PhysicalMediaInLocation association. In addition, one
can determine if a Location is empty or full based on
whether this association exists for the
StorageMediaLocation.'
```

```
SUP dlm1Dependency ABSTRACT
MAY (dlmOrientation $ dlmTimeOfLastPlacement)
)

(1.3.6.1.4.1.412.100.3.2.1121 NAME
'dlmPhysicalMediaInLocationAntecedentRef'
DESC 'The StorageMediaLocation which holds the
PhysicalMedia. The value of this attribute points to an
entry of class dlmStorageMediaLocation.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1122 NAME
'dlmPhysicalMediaInLocationDependentRef'
DESC 'The Media in the Location. Values of this
attribute point to entries of class
dlmPhysicalMedia.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.85 NAME
'dlm1PhysicalMediaInLocationAuxClass'
DESC 'Within a StorageLibrary, all Media should be
accounted for, and be present in some Storage
Location. This relationship is made explicit by the
PhysicalMediaInLocation association. In addition, one
can determine if a Location is empty or full based on
whether this association exists for the
```

```

 StorageMediaLocation.'
 SUP dlm1PhysicalMediaInLocation AUXILIARY
 MAY (dlmPhysicalMediaInLocationAntecedentRef $
 dlmPhysicalMediaInLocationDependentRef)
)

```

### 3.33 ElementsLinked Classes

These classes indicate which PhysicalElements are cabled together by a PhysicalLink.

```

(1.3.6.1.4.1.412.100.3.1.2.86 NAME 'dlm1ElementsLinked'
 DESC 'The ElementsLinked association indicates which
 Physical Elements are cabled together by a
 PhysicalLink.'
 SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1123 NAME 'dlmElementsLinkedAntecedentRef'
 DESC 'The PhysicalLink. Values of this attribute
 point to entries of class dlmPhysicalLink.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1124 NAME 'dlmElementsLinkedDependentRef'
 DESC 'The PhysicalElement that is linked. Values of
 this attribute points to entries of class
 dlmPhysicalElement.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.87 NAME 'dlm1ElementsLinkedAuxClass'
 DESC 'The ElementsLinked association indicates which
 Physical Elements are cabled together by a
 PhysicalLink.'
 SUP dlm1ElementsLinked AUXILIARY
 MAY (dlmElementsLinkedAntecedentRef $
 dlmElementsLinkedDependentRef)
)

```

### 3.34 ConnectedTo Classes

These classes indicate that two or more PhysicalConnectors are connected together.

```

(1.3.6.1.4.1.412.100.3.1.2.88 NAME 'dlm1ConnectedTo'
 DESC 'The ConnectedTo association indicates that two or
 more PhysicalConnectors are connected together.'
 SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1125 NAME 'dlmConnectedToAntecedentRef'
 DESC 'The Antecedent represents a PhysicalConnector
 that serves as one end of the connection. Values of

```



```

 this attribute points to entries of class
 dlmPhysicalConnector.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1126 NAME 'dlmConnectedToDependentRef'
DESC 'The Dependent represents another
PhysicalConnector that serves as the other end of the
connection. Values of this attribute point to
entries of class dlmPhysicalConnector.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.89 NAME 'dlm1ConnectedToAuxClass'
DESC 'The ConnectedTo association indicates that two or
more PhysicalConnectors are connected together.'
SUP dlm1ConnectedTo AUXILIARY
MAY (dlmConnectedToAntecedentRef $
dlmConnectedToDependentRef)
)

```

### 3.35 SlotInSlot Classes

These classes represent the ability of a special adapter to extend an existing Slot structure to enable otherwise incompatible Cards.

```

(1.3.6.1.4.1.412.100.3.1.2.90 NAME 'dlm1SlotInSlot'
DESC 'Slots are special types of Connectors into which
adapter Cards are typically inserted. The SlotInSlot
relationship represents the ability of a special
adapter to extend the existing Slot structure to
enable otherwise incompatible Cards to be plugged into
a Frame or Hosting Board. The adapter effectively
creates a new Slot and can be thought of
(conceptually) as a Slot in a Slot. This enables Cards
that would otherwise be physically and/or electrically
incompatible with the existing Slots to be supported,
by interfacing to the Slot provided by the adapter.
This has many practical uses. For example, networking
boards are very expensive. As new hardware becomes
available, Chassis and even Card configurations
change. To protect the investment of their customers,
networking vendors will manufacture special adapters
that enable old Cards to fit into new Chassis or
Hosting Boards and/or new Cards to fit into old. This
is done using a special adapter that fits over one or
more existing Slots and presents a new Slot into which
the Card can plug.'
SUP dlm1ConnectedTo ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1127 NAME 'dlmSlotInSlotAntecedentRef'
DESC 'The Antecedent represents the existing Slot(s) of
the HostingBoard, or Frame that are being adapted to
accommodate a Card that would otherwise not be

```

```

 physically and/or electrically compatible with it.
 Values of this attribute point to entries of class
 dlmSlot.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1128 NAME 'dlmSlotInSlotDependentRef'
DESC 'The new Slot provided by the adapter board.
The value of this attribute points to an entry of class
dlmSlot.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.91 NAME 'dlm1SlotInSlotAuxClass'
DESC 'Slots are special types of Connectors into which
adapter Cards are typically inserted. The SlotInSlot
relationship represents the ability of a special
adapter to extend the existing Slot structure to
enable otherwise incompatible Cards to be plugged into
a Frame or Hosting Board. The adapter effectively
creates a new Slot and can be thought of
(conceptually) as a Slot in a Slot. This enables Cards
that would otherwise be physically and/or electrically
incompatible with the existing Slots to be supported,
by interfacing to the Slot provided by the adapter.
This has many practical uses. For example, networking
boards are very expensive. As new hardware becomes
available, Chassis and even Card configurations
change. To protect the investment of their customers,
networking vendors will manufacture special adapters
that enable old Cards to fit into new Chassis or
Hosting Boards and/or new Cards to fit into old. This
is done using a special adapter that fits over one or
more existing Slots and presents a new Slot into which
the Card can plug.'
SUP dlm1SlotInSlot AUXILIARY
MAY (dlmSlotInSlotAntecedentRef $
dlmSlotInSlotDependentRef)
)

```

### 3.36 PackageInConnector Classes

These classes define the relationship between Packages and SystemConnectors that provide power and/or transfer data.

```

(1.3.6.1.4.1.412.100.3.1.2.92 NAME 'dlm1PackageInConnector'
DESC 'Adapter cards and other "packaging" are plugged
into System Connectors for power and/or to transfer
data. This relationship is defined by
PackageInConnector. For example, it would be used to
describe the insertion of a daughtercard onto another
Card. Various subclasses of PackageInConnector are
also defined. PackageInSlot and its subclass,
CardInSlot, are two examples of subclasses.'
SUP dlm1Dependency ABSTRACT
)

```

```

)

(1.3.6.1.4.1.412.100.3.2.1129 NAME
'dlmPackageInConnectorAntecedentRef'
 DESC 'The Connector into which the Package is inserted.
 Values of this attribute point to entries of class
 dlmPhysicalConnector.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1130 NAME
'dlmPackageInConnectorDependentRef'
 DESC 'The Package in the Connector. The value of this
 attribute points to an entry of class
 dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.93 NAME 'dlm1PackageInConnectorAuxClass'
 DESC 'Adapter cards and other "packaging" are plugged
 into System Connectors for power and/or to transfer
 data. This relationship is defined by
 PackageInConnector. For example, it would be used to
 describe the insertion of a daughtercard onto another
 Card. Various subclasses of PackageInConnector are
 also defined. PackageInSlot and its subclass,
 CardInSlot, are two examples of subclasses.'
 SUP dlm1PackageInConnector AUXILIARY
 MAY (dlmPackageInConnectorAntecedentRef $
 dlmPackageInConnectorDependentRef)
)

```

### 3.37 PackageInSlot Classes

These classes model the enhancement of Chassis to support addition Chassis devices.

```

(1.3.6.1.4.1.412.100.3.1.2.94 NAME 'dlm1PackageInSlot'
 DESC 'Complex networking devices often are
 Chassis-based. These Chassis allow for enhancement
 and/or augmentation of their base functionality by
 accepting additional Chassis devices, similar to
 accepting functionality in the form of adding Cards.
 This association models this capability.'
 SUP dlm1PackageInConnector ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1131 NAME 'dlmPackageInSlotAntecedentRef'
 DESC 'The Slot into which the PhysicalPackage is
 inserted. Values of this attribute point to
 entries of class dlmSlot.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

```

```
(1.3.6.1.4.1.412.100.3.2.1132 NAME 'dmlPackageInSlotDependentRef'
 DESC 'The Package in the Slot. The value of this
 attribute points to an entry of class
 dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.95 NAME 'dml1PackageInSlotAuxClass'
 DESC 'Complex networking devices often are
 Chassis-based. These Chassis allow for enhancement
 and/or augmentation of their base functionality by
 accepting additional Chassis devices, similar to
 accepting functionality in the form of adding Cards.
 This association models this capability.'
 SUP dml1PackageInSlot AUXILIARY
 MAY (dmlPackageInSlotAntecedentRef $
 dmlPackageInSlotDependentRef)
)
```

### 3.38 CardInSlot Classes

These classes make explicit the relationship of a Card inserted into a Slot.

```
(1.3.6.1.4.1.412.100.3.1.2.96 NAME 'dml1CardInSlot'
 DESC 'Slots are special types of Connectors into which
 adapter Cards are inserted. This relationship of a
 Card in a Slot is made explicit using the CardInSlot
 association.'
 SUP dml1PackageInSlot ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1133 NAME 'dmlCardInSlotAntecedentRef'
 DESC 'The Slot into which the Card is inserted.
 Values of this attribute point to entries of class
 dmlSlot.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1134 NAME 'dmlCardInSlotDependentRef'
 DESC 'The Card in the Slot. The value of this attribute
 points to an entry of class dmlCard.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.97 NAME 'dml1CardInSlotAuxClass'
 DESC 'Slots are special types of Connectors into which
 adapter Cards are inserted. This relationship of a
 Card in a Slot is made explicit using the CardInSlot
 association.'
 SUP dml1CardInSlot AUXILIARY
 MAY (dmlCardInSlotAntecedentRef $
 dmlCardInSlotDependentRef)
)
```

### 3.39 AdapterActiveConnection Classes

These classes indicate that a NetworkAdapter is using the referenced PhysicalConnector to output to the network.

```
(1.3.6.1.4.1.412.100.3.1.2.98 NAME 'dlm1AdapterActiveConnection'
DESC 'The AdapterActiveConnection relationship
 indicates that a NetworkAdapter is using the
 referenced PhysicalConnector to output to the network.
 This relationship is important when the Adapter can
 choose to output from one of several Connectors. The
 Connectors may be associated with the NetworkAdapter
 in a Realizes relationship - but this is not required.
 This association provides additional information (i.e.,
 "in use for communication") different than Realizes.'
SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1135 NAME
'dlmAdapterActiveConnectionAntecedentRef'
DESC 'The PhysicalConnector. Values of this attribute
 point to entries of class dlmPhysicalConnector.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1136 NAME
'dlmAdapterActiveConnectionDependentRef'
DESC 'The NetworkAdapter that transmits using the
 Connector. Values of this attribute point to
 entries of class dlmNetworkAdapter.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.99 NAME
'dlm1AdapterActiveConnectionAuxClass'
DESC 'The AdapterActiveConnection relationship
 indicates that a NetworkAdapter is using the
 referenced PhysicalConnector to output to the network.
 This relationship is important when the Adapter can
 choose to output from one of several Connectors. The
 Connectors may be associated with the NetworkAdapter
 in a Realizes relationship - but this is not required.
 This association provides additional information (i.e.,
 "in use for communication") different than Realizes.'
SUP dlm1AdapterActiveConnection AUXILIARY
MAY (dlmAdapterActiveConnectionAntecedentRef $
 dlmAdapterActiveConnectionDependentRef)
)
```

### 3.40 ComputerSystemPackage Classes

These classes explicitly define the relationship between the UnitaryComputerSystem that is realized by one or more PhysicalPackages.

## DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
(1.3.6.1.4.1.412.100.3.2.1137 NAME 'dlmPlatformGUID'
 DESC 'A Globally Unique Identifier for the System's
 Package.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 EQUALITY caseIgnoreMatch
)

(1.3.6.1.4.1.412.100.3.1.2.100 NAME 'dlm1ComputerSystemPackage'
 DESC 'Similar to the way that LogicalDevices are
 "Realized" by PhysicalElements, UnitaryComputerSystems
 are realized in one or more PhysicalPackages. The
 ComputerSystemPackage association explicitly defines
 this relationship.'
 SUP dlm1Dependency ABSTRACT
 MAY (dlmPlatformGUID)
)

(1.3.6.1.4.1.412.100.3.2.1138 NAME
'dlmComputerSystemPackageAntecedentRef'
 DESC 'The PhysicalPackage(s) that realize a
 UnitaryComputerSystem. The value of this attribute
 points to an entry of class dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1139 NAME
'dlmComputerSystemPackageDependentRef'
 DESC 'The UnitaryComputerSystem. Values of this
 attribute point to entries of class
 dlmUnitaryComputerSystem.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.101 NAME
'dlm1ComputerSystemPackageInstance'
 DESC 'Similar to the way that LogicalDevices are
 "Realized" by PhysicalElements, UnitaryComputerSystems
 are realized in one or more PhysicalPackages. The
 ComputerSystemPackage association explicitly defines
 this relationship.'
 SUP dlm1ComputerSystemPackage
 MAY (dlmComputerSystemPackageAntecedentRef $
 dlmComputerSystemPackageDependentRef)
)

(1.3.6.1.4.1.412.100.3.3.2.23 NAME
'dlm1ComputerSystemPackageInstanceNameForm'
 OC dlm1ComputerSystemPackageInstance
 MUST (orderedCimKeys)
)

(<physical-sr-23> NAME
'dlm1ComputerSystemPackageInstanceStructureRule'
 Form dlm1ComputerSystemPackageInstanceNameForm
)
```

```
(1.3.6.1.4.1.412.100.3.2.1140 NAME
'dlmComputerSystemPackageHelperRef'
 DESC 'Pointer to ComputerSystemPackageInstance.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.102 NAME
'dlm1ComputerSystemPackageHelperAuxClass'
 DESC 'Helper class for finding ComputerSystemPackage.'
 SUP top AUXILIARY
 MAY (dlmComputerSystemPackageHelperRef)
)
```

### 3.41 LibraryPackage Classes

The **LibraryPackage** classes explicitly define this realization of a **StorageLibrary** by one or more **PhysicalPackages**.

```
(1.3.6.1.4.1.412.100.3.1.2.103 NAME 'dlm1LibraryPackage'
 DESC 'Similar to the way that LogicalDevices are
 "Realized" by PhysicalElements, a StorageLibrary can
 be realized in one or more PhysicalPackages. The
 LibraryPackage association explicitly defines this
 relationship.'
 SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1141 NAME 'dlmLibraryPackageAntecedentRef'
 DESC 'The PhysicalPackage(s) that realize a
 StorageLibrary. Values of this attribute point to
 entries of class dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1142 NAME 'dlmLibraryPackageDependentRef'
 DESC 'The StorageLibrary. Values of this attribute
 point to entries of class dlmStorageLibrary.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.104 NAME 'dlm1LibraryPackageAuxClass'
 DESC 'Similar to the way that LogicalDevices are
 "Realized" by PhysicalElements, a StorageLibrary can
 be realized in one or more PhysicalPackages. The
 LibraryPackage association explicitly defines this
 relationship.'
 SUP dlm1LibraryPackage AUXILIARY
 MAY (dlmLibraryPackageAntecedentRef $
 dlmLibraryPackageDependentRef)
)
```

### 3.42 PackageCooling Classes

**These classes model the association between a Package and a CoolingDevice installed to assist the cooling of the Package in general.**

```
(1.3.6.1.4.1.412.100.3.1.2.105 NAME 'dlm1PackageCooling'
DESC 'Often, a CoolingDevice is installed in a Package
 such as a Chassis or a Rack, not for a specific
 Device, but to assist in the cooling of the Package in
 general. This relationship is described by the
 PackageCooling association.'
SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1143 NAME 'dlmPackageCoolingAntecedentRef'
DESC 'The CoolingDevice for the Package. Values of
 this attribute point to entries of class
 dlmCoolingDevice.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1144 NAME 'dlmPackageCoolingDependentRef'
DESC 'The PhysicalPackage whose environment is cooled.
 Values of this attribute point to entries of class
 dlmPhysicalPackage.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.106 NAME 'dlm1PackageCoolingAuxClass'
DESC 'Often, a CoolingDevice is installed in a Package
 such as a Chassis or a Rack, not for a specific
 Device, but to assist in the cooling of the Package in
 general. This relationship is described by the
 PackageCooling association.'
SUP dlm1PackageCooling AUXILIARY
MAY (dlmPackageCoolingAntecedentRef $
 dlmPackageCoolingDependentRef)
)
```

### 3.43 PackageTempSensor Classes

**These classes describe the relationship between a Package and the TemperatureSensor installed to measure the Package environment in general.**

```
(1.3.6.1.4.1.412.100.3.1.2.107 NAME 'dlm1PackageTempSensor'
DESC 'Often, a TemperatureSensor is installed in a
 Package such as a Chassis or a Rack, not to measure
 any particular Device, but the Package's environment
 in general. This relationship is described by the
 PackageTempSensor association.'
SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1145 NAME
'dlmPackageTempSensorAntecedentRef'
DESC 'The TemperatureSensor for the Package. Values
```



```

 of this attribute point to entries of class
 dlmTemperatureSensor.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1146 NAME
'dlmPackageTempSensorDependentRef'
 DESC 'The PhysicalPackage whose environment is
 monitored. Values of this attribute point to
 entries of class dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.108 NAME 'dlm1PackageTempSensorAuxClass'
 DESC 'Often, a TemperatureSensor is installed in a
 Package such as a Chassis or a Rack, not to measure
 any particular Device, but the Package"s environment
 in general. This relationship is described by the
 PackageTempSensor association.'
 SUP dlm1PackageTempSensor AUXILIARY
 MAY (dlmPackageTempSensorAntecedentRef $
 dlmPackageTempSensorDependentRef)
)

```

### 3.44 PackageAlarm Classes

These classes describe the relationship between a Package and the AlarmDevice installed to indicate issues with the Package's environment, security state, or general health.

```

(1.3.6.1.4.1.412.100.3.1.2.109 NAME 'dlm1PackageAlarm'
 DESC 'Often, an AlarmDevice is installed as part of a
 Package, not to indicate issues with any particular
 LogicalDevice or PhysicalComponent, but with the
 Package"s environment in general, its security state
 or its overall health. This relationship is described
 by the PackageAlarm association. '
 SUP dlm1Dependency ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1147 NAME 'dlmPackageAlarmAntecedentRef'
 DESC 'The AlarmDevice for the Package. . Values of
 this attribute point to entries of class
 dlmAlarmDevice.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1148 NAME 'dlmPackageAlarmDependentRef'
 DESC 'The PhysicalPackage whose health, security,
 environment, etc. is alarmed. . Values of this
 attribute point to entries of class
 dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

```

```

)
(1.3.6.1.4.1.412.100.3.1.2.110 NAME 'dlm1PackageAlarmAuxClass'
 DESC 'Often, an AlarmDevice is installed as part of a
 Package, not to indicate issues with any particular
 LogicalDevice or PhysicalComponent, but with the
 Package"s environment in general, its security state
 or its overall health. This relationship is described
 by the PackageAlarm association. '
 SUP dlm1PackageAlarm AUXILIARY
 MAY (dlmPackageAlarmAntecedentRef $
 dlmPackageAlarmDependentRef)
)

```

### 3.45 HomeForMedia Classes

These classes express the relationship between a specific Media entry and its "home" Location entry.

```

(1.3.6.1.4.1.412.100.3.1.2.111 NAME 'dlm1HomeForMedia'
 DESC ' In a StorageLibrary or repository for
 PhysicalMedia, it is reasonable that a specific Media
 always be placed in a specific Location (rather than
 anywhere in the repository/Library). The default
 location for a PhysicalMedia is called its "home".
 This information is expressed by the HomeForMedia
 dependency association. (Note that a PhysicalMedia can
 have at most one "home", and a StorageMediaLocation can
 be "home" to at most one Media.) '
 SUP dlm1Dependency ABSTRACT
)
(1.3.6.1.4.1.412.100.3.2.1149 NAME 'dlmHomeForMediaAntecedentRef'
 DESC 'The Location that is "home" to a piece of
 PhysicalMedia. The value of this attribute points to
 an entry of class dlmStorageMediaLocation.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)
(1.3.6.1.4.1.412.100.3.2.1150 NAME 'dlmHomeForMediaDependentRef'
 DESC 'The PhysicalMedia whose "home" or default
 Location is defined. The value of this attribute points
 to an entry of class dlmPhysicalMedia.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)
(1.3.6.1.4.1.412.100.3.1.2.112 NAME 'dlm1HomeForMediaAuxClass'
 DESC ' In a StorageLibrary or repository for
 PhysicalMedia, it is reasonable that a specific Media
 always be placed in a specific Location (rather than
 anywhere in the repository/Library). The default
 location for a PhysicalMedia is called its "home".
 This information is expressed by the HomeForMedia
 dependency association. (Note that a PhysicalMedia can
 have at most one "home", and a StorageMediaLocation can

```

```

 be "home" to at most one Media.)'
 SUP dlm1HomeForMedia AUXILIARY
 MAY (dlmHomeForMediaAntecedentRef $
 dlmHomeForMediaDependentRef)
)

```

### 3.46 Container Classes

The Container Classes model the relationship between a contained and containing PhysicalElement.

```

(1.3.6.1.4.1.412.100.3.2.1151 NAME 'dlmLocationWithinContainer'
 DESC 'A free-form string representing the positioning
 of the PhysicalElement within the PhysicalPackage.
 Information relative to stationary elements in the
 Container (for example, "second drive bay from the
 top"), angles, altitudes and other data may be
 recorded in this property. This string could
 supplement or be used in place of instantiating the
 Location object.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 EQUALITY caseIgnoreMatch
)

(1.3.6.1.4.1.412.100.3.1.2.113 NAME 'dlm1Container'
 DESC 'The Container association represents the
 relationship between a contained and a containing
 PhysicalElement. A containing object must be a
 PhysicalPackage.'
 SUP dlm1Component ABSTRACT
 MAY (dlmLocationWithinContainer)
)

(1.3.6.1.4.1.412.100.3.2.1152 NAME 'dlmContainerGroupComponentRef'
 DESC 'The PhysicalPackage that contains other
 PhysicalElements, including other Packages. The value
 of this attribute points to an entry of class
 dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1153 NAME 'dlmContainerPartComponentRef'
 DESC 'The PhysicalElement which is contained in the
 Package. Values of this attribute point to entries
 of class dlmPhysicalElement.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.114 NAME 'dlm1ContainerAuxClass'
 DESC 'The Container association represents the
 relationship between a contained and a containing
 PhysicalElement. A containing object must be a
 PhysicalPackage.'
 SUP dlm1Container AUXILIARY
 MAY (dlmContainerGroupComponentRef $

```

```
 dlmContainerPartComponentRef)
)
```

### 3.47 ChassisInRack Classes

These classes model the "containing" relationship between a Rack and the Chassis it contains.

```
(1.3.6.1.4.1.412.100.3.2.1154 NAME 'dlmBottomU'
 DESC 'An integer indicating the lowest or "bottom" U in
 which the Chassis is mounted. A "U" is a standard unit
 of measure for the height of a Rack or rack-mountable
 component. It is equal to 1.75 inches or 4.445 cm. The
 value is considered to be Us.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
 EQUALITY integerMatch
)
```

```
(1.3.6.1.4.1.412.100.3.1.2.115 NAME 'dlm1ChassisInRack'
 DESC 'Racks, as simple enclosures, contain Chassis that
 provide the physical componentry realizing processing
 nodes, storage devices, UPSs, etc. The ChassisInRack
 association makes explicit the "containing"
 relationship between the Rack and the Chassis.'
 SUP dlm1Container ABSTRACT
 MAY (dlmBottomU)
)
```

```
(1.3.6.1.4.1.412.100.3.2.1155 NAME
'dlmChassisInRackGroupComponentRef'
 DESC 'The Rack that contains the Chassis. The value of
 this attribute points to an entry of class dlmRack.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)
```

```
(1.3.6.1.4.1.412.100.3.2.1156 NAME
'dlmChassisInRackPartComponentRef'
 DESC 'The Chassis which is mounted in the Rack.
 Values of this attribute point to entries of class
 dlmChassis.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)
```

```
(1.3.6.1.4.1.412.100.3.1.2.116 NAME 'dlm1ChassisInRackAuxClass'
 DESC 'Racks, as simple enclosures, contain Chassis that
 provide the physical componentry realizing processing
 nodes, storage devices, UPSs, etc. The ChassisInRack
 association makes explicit the "containing"
 relationship between the Rack and the Chassis.'
 SUP dlm1ChassisInRack AUXILIARY
 MAY (dlmChassisInRackGroupComponentRef $
 dlmChassisInRackPartComponentRef)
)
```

### 3.48 PackageInChassis Classes

### These classes describe the containment of Packages in a Chassis.

```
(1.3.6.1.4.1.412.100.3.1.2.117 NAME 'dlm1PackageInChassis'
 DESC 'A Chassis can contain other Packages, such as
 other Chassis and Cards. The PackageInChassis
 association makes explicit this relationship.'
 SUP dlm1Container ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1157 NAME
'dlmPackageInChassisGroupComponentRef'
 DESC 'The Chassis that contains other PhysicalPackages.
 The value of this attribute points to an entry of class
 dlmChassis.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1158 NAME
'dlmPackageInChassisPartComponentRef'
 DESC 'The PhysicalPackage which is contained in the
 Chassis. Values of this attribute point to entries
 of class dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.118 NAME 'dlm1PackageInChassisAuxClass'
 DESC 'A Chassis can contain other Packages, such as
 other Chassis and Cards. The PackageInChassis
 association makes explicit this relationship.'
 SUP dlm1PackageInChassis AUXILIARY
 MAY (dlmPackageInChassisGroupComponentRef $
 dlmPackageInChassisPartComponentRef)
)
```

### 3.49 CardOnCard Classes

#### These classes model the plugging in of Cards or Card-like modules into Motherboards, baseboards, or other Cards.

```
(1.3.6.1.4.1.412.100.3.2.1159 NAME 'dlmMountOrSlotDescription'
 DESC 'A string describing and identifying how the Card
 is mounted on or plugged into the "other" Card. Slot
 information could be included in this field and may be
 sufficient for certain management purposes. If so, this
 avoids creating instantiations of Connector/Slot
 objects just to model the relationship of Cards to
 HostingBoards or other adapters. On the other hand, if
 Slot and Connector information is available, this field
 could be used to provide more detailed mounting or slot
 insertion data.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 EQUALITY caseIgnoreMatch
)

(1.3.6.1.4.1.412.100.3.1.2.119 NAME 'dlm1CardOnCard'
```

```

DESC 'Cards may be plugged into
 Motherboards/baseboards, are daughtercards of an
 adapter, or support special Card-like modules. These
 relationships are described by the CardOnCard
 association.'
SUP dlm1Container ABSTRACT
MAY (dlmMountOrSlotDescription)
)

(1.3.6.1.4.1.412.100.3.2.1160 NAME 'dlmCardOnCardGroupComponentRef'
DESC 'The Card that hosts another Card. The value of
 this attribute points to an entry of class dlmCard.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1161 NAME 'dlmCardOnCardPartComponentRef'
DESC 'The Card that is plugged into or otherwise
 mounted on another Card. Values of this attribute
 point to entries of class dlmCard.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.120 NAME 'dlm1CardOnCardAuxClass'
DESC 'Cards may be plugged into
 Motherboards/baseboards, are daughtercards of an
 adapter, or support special Card-like modules. These
 relationships are described by the CardOnCard
 association.'
SUP dlm1CardOnCard AUXILIARY
MAY (dlmCardOnCardGroupComponentRef $
 dlmCardOnCardPartComponentRef)
)

```

### 3.50 PackagedComponent Classes

These classes make the containment of a Component in a PhysicalPackage explicit. This is because a component may be removed from or not yet inserted into its containing Package

```

(1.3.6.1.4.1.412.100.3.1.2.121 NAME 'dlm1PackagedComponent'
DESC 'A Component is typically contained by a
 PhysicalPackage, such as a Chassis or Card. The
 PackagedComponent association makes this relationship
 explicit. In the first sentence, the word,
 "typically", is used. This is because a Component may
 be removed from, or not yet inserted into, its
 containing Package (ie, the Removable boolean is
 TRUE). Therefore, a Component may not always be
 associated with a container.'
SUP dlm1Container ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1162 NAME
'dlmPackagedComponentGroupComponentRef'
DESC 'The PhysicalPackage that contains Component(s).

```

```

 The value of this attribute points to an entry of class
 dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1163 NAME
'dlmPackagedComponentPartComponentRef'
 DESC 'The PhysicalComponent which is contained in the
 Package. Values of this attribute point to entries
 of class dlmPhysicalComponent.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.122 NAME 'dlm1PackagedComponentAuxClass'
 DESC 'A Component is typically contained by a
 PhysicalPackage, such as a Chassis or Card. The
 PackagedComponent association makes this relationship
 explicit. In the first sentence, the word,
 "typically", is used. This is because a Component may
 be removed from, or not yet inserted into, its
 containing Package (ie, the Removable boolean is
 TRUE). Therefore, a Component may not always be
 associated with a container.'
 SUP dlm1PackagedComponent AUXILIARY
 MAY (dlmPackagedComponentGroupComponentRef $
 dlmPackagedComponentPartComponentRef)
)

```

### 3.51 MemoryOnCard Classes

These classes make explicit the location of PhysicalMemory on cards.

```

(1.3.6.1.4.1.412.100.3.1.2.123 NAME 'dlm1MemoryOnCard'
 DESC 'PhysicalMemory can be located on HostingBoards,
 adapter Cards, etc. This association explicitly
 defines this relationship of Memory to Cards.'
 SUP dlm1PackagedComponent ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1164 NAME
'dlmMemoryOnCardGroupComponentRef'
 DESC 'The Card that includes or "contains" Memory.
 The value of this attribute points to an entry of class
 dlmCard.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1165 NAME 'dlmMemoryOnCardPartComponentRef'
 DESC 'The PhysicalMemory which is located on the Card.
 Values of this attribute points to entries of class
 dlmPhysicalMemory.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

```

```
(1.3.6.1.4.1.412.100.3.1.2.124 NAME 'dlm1MemoryOnCardAuxClass'
 DESC 'PhysicalMemory can be located on HostingBoards,
 adapter Cards, etc. This association explicitly
 defines this relationship of Memory to Cards.'
 SUP dlm1MemoryOnCard AUXILIARY
 MAY (dlmMemoryOnCardGroupComponentRef $
 dlmMemoryOnCardPartComponentRef)
)
```

### 3.52 ConnectorOnPackage Classes

These classes describe the containment relationship between PhysicalPackages and the Connectors they contain.

```
(1.3.6.1.4.1.412.100.3.1.2.125 NAME 'dlm1ConnectorOnPackage'
 DESC 'PhysicalPackages contain Connectors as well as
 other PhysicalElements. The ConnectorOnPackage
 association makes explicit the containment
 relationship between Connectors and Packages.'
 SUP dlm1Container ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1166 NAME
'dlmConnectorOnPackageGroupComponentRef'
 DESC 'The PhysicalPackage that has a Connector.
 The value of this attribute points to an entry of class
 dlmPhysicalPackage.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1167 NAME
'dlmConnectorOnPackagePartComponentRef'
 DESC 'The PhysicalConnector. Values of this attribute
 point to entries of class dlmPhysicalConnector.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.126 NAME 'dlm1ConnectorOnPackageAuxClass'
 DESC 'PhysicalPackages contain Connectors as well as
 other PhysicalElements. The ConnectorOnPackage
 association makes explicit the containment
 relationship between Connectors and Packages.'
 SUP dlm1ConnectorOnPackage AUXILIARY
 MAY (dlmConnectorOnPackageGroupComponentRef $
 dlmConnectorOnPackagePartComponentRef)
)
```

### 3.53 LinkHasConnector Classes

These classes define the PhysicalConnectors that a Cable or Link uses to actually connect PhysicalElements.

```
(1.3.6.1.4.1.412.100.3.1.2.127 NAME 'dlm1LinkHasConnector'
 DESC 'Cables and Links utilize PhysicalConnectors to
```



```

 actually "connect" PhysicalElements. This association
 explicitly defines this relationship of Connectors for
 PhysicalLinks.'
 SUP dlm1Component ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1168 NAME
'dlmLinkHasConnectorGroupComponentRef'
 DESC 'The PhysicalLink that has a Connector. The value
 of this attribute points to an entry of class
 dlmPhysicalLink.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1169 NAME
'dlmLinkHasConnectorPartComponentRef'
 DESC 'The PhysicalConnector. Values of this attribute
 point to entries of class dlmPhysicalConnector.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.128 NAME 'dlm1LinkHasConnectorAuxClass'
 DESC 'Cables and Links utilize PhysicalConnectors to
 actually "connect" PhysicalElements. This association
 explicitly defines this relationship of Connectors for
 PhysicalLinks.'
 SUP dlm1LinkHasConnector AUXILIARY
 MAY (dlmLinkHasConnectorGroupComponentRef $
 dlmLinkHasConnectorPartComponentRef)
)

```

### 3.54 PhysicalElementLocation Classes

**These classes associate a PhysicalElement with a Location object for inventory or replacement purposes.**

```

(1.3.6.1.4.1.412.100.3.1.2.129 NAME 'dlm1PhysicalElementLocation'
 DESC 'PhysicalElementLocation associates a
 PhysicalElement with a Location object for inventory
 or replacement purposes.'
 SUP top ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1170 NAME
'dlmPhysicalElementLocationElementRef'
 DESC 'The PhysicalElement whose Location is specified.
 Values of this attribute point to entries of class
 dlmPhysicalElement.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1171 NAME
'dlmPhysicalElementLocationPhysicalLocationRef'
 DESC 'The PhysicalElement"s Location. The value of this

```

```

 attribute points to an entry of class dlmLocation.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.130 NAME
'dlm1PhysicalElementLocationAuxClass'
DESC 'PhysicalElementLocation associates a
PhysicalElement with a Location object for inventory
or replacement purposes.'
SUP dlm1PhysicalElementLocation AUXILIARY
MAY (dlmPhysicalElementLocationElementRef $
 dlmPhysicalElementLocationPhysicalLocationRef)
)

```

### 3.55 ElementCapacity Classes

These classes associate a PhysicalCapacity entry with one or more PhysicalElements.

```

(1.3.6.1.4.1.412.100.3.1.2.131 NAME 'dlm1ElementCapacity'
DESC 'ElementCapacity associates a PhysicalCapacity
object with one or more PhysicalElements. It serves
to associate a description of min/max hardware
requirements or capabilities (stored as a kind of
PhysicalCapacity), with the PhysicalElements being
described.'
SUP top ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1172 NAME 'dlmElementCapacityCapacityRef'
DESC 'PhysicalCapacity describes the minimum and
maximum requirements, and ability to support different
types of hardware for a PhysicalElement. Values of
this attribute point to entries of class
dlmPhysicalCapacity.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1173 NAME 'dlmElementCapacityElementRef'
DESC 'The PhysicalElement being described. Values of
this attribute point to entries of class
dlmPhysicalElement.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.132 NAME 'dlm1ElementCapacityAuxClass'
DESC 'ElementCapacity associates a PhysicalCapacity
object with one or more PhysicalElements. It serves
to associate a description of min/max hardware
requirements or capabilities (stored as a kind of
PhysicalCapacity), with the PhysicalElements being
described.'
SUP dlm1ElementCapacity AUXILIARY
MAY (dlmElementCapacityCapacityRef $

```

```
 dlmElementCapacityElementRef)
)
```

### 3.56 ParticipatesInSet Classes

These classes indicate which PhysicalElements should be replaced together.

```
(1.3.6.1.4.1.412.100.3.1.2.133 NAME 'dlm1ParticipatesInSet'
 DESC 'ParticipatesInSet indicates which
 PhysicalElements should be replaced together.'
 SUP top ABSTRACT
)

(1.3.6.1.4.1.412.100.3.2.1174 NAME 'dlmParticipatesInSetElementRef'
 DESC 'The PhysicalElement which should be replaced with
 other Elements, as a Set. Values of this attribute
 point to entries of class dlmPhysicalElement.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1175 NAME 'dlmParticipatesInSetSetRef'
 DESC 'The ReplacementSet. Values of this attribute
 point to entries of class dlmReplacementSet.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.134 NAME 'dlm1ParticipatesInSetAuxClass'
 DESC 'ParticipatesInSet indicates which
 PhysicalElements should be replaced together.'
 SUP dlm1ParticipatesInSet AUXILIARY
 MAY (dlmParticipatesInSetElementRef $
 dlmParticipatesInSetSetRef)
)
```

### 3.57 AdjacentSlots Classes

These classes describe the layout of Slots on a HostingBoard or adapter Card.

```
(1.3.6.1.4.1.412.100.3.2.1176 NAME 'dlmDistanceBetweenSlots'
 DESC 'The distance, in inches, between adjacent Slots.
 The value is considered to be Inches.'
 SUP cimFloat32 SINGLE-VALUE
)

(1.3.6.1.4.1.412.100.3.2.1177 NAME 'dlmSharedSlots'
 DESC 'Slots can be located in close proximity on
 Hosting Boards or other Cards, such that if one of
 these Slots is populated by an adapter Card, the other
 Slot must be left empty. This relationship is indicated
 by the SharedSlots boolean set to TRUE.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE
)
```

## DMTF LDAP Schema for the CIM v2.5 Physical Information Model

```
(1.3.6.1.4.1.412.100.3.1.2.135 NAME 'dlm1AdjacentSlots'
 DESC 'AdjacentSlots describes the layout of Slots on a
 HostingBoard or adapter Card. Information like the
 distance between the Slots and whether they are
 "shared" (if one is populated, then the other Slot can
 not be used), is conveyed as properties of the
 association.'
 SUP top ABSTRACT
 MAY (dlmDistanceBetweenSlots $ dlmSharedSlots)
)

(1.3.6.1.4.1.412.100.3.2.1178 NAME 'dlmAdjacentSlotsSlotARef'
 DESC 'One of the adjacent Slots. The value of this
 attribute points to an entry of class dlmSlot.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.2.1179 NAME 'dlmAdjacentSlotsSlotBRef'
 DESC 'The "other" adjacent Slot. The value of this
 attribute points to an entry of class dlmSlot.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.136 NAME 'dlm1AdjacentSlotsInstance'
 DESC 'AdjacentSlots describes the layout of Slots on a
 HostingBoard or adapter Card. Information like the
 distance between the Slots and whether they are
 "shared" (if one is populated, then the other Slot can
 not be used), is conveyed as properties of the
 association.'
 SUP dlm1AdjacentSlots
 MAY (dlmAdjacentSlotsSlotARef $
 dlmAdjacentSlotsSlotBRef)
)

(1.3.6.1.4.1.412.100.3.3.2.24 NAME
'dlm1AdjacentSlotsInstanceNameForm'
 OC dlm1AdjacentSlotsInstance
 MUST (orderedCimKeys)
)

(<physical-sr-24> NAME 'dlm1AdjacentSlotsInstanceStructureRule'
 Form dlm1AdjacentSlotsInstanceNameForm
)

(1.3.6.1.4.1.412.100.3.2.1180 NAME 'dlmAdjacentSlotsHelperRef'
 DESC 'Pointer to AdjacentSlotsInstance.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 EQUALITY distinguishedNameMatch
)

(1.3.6.1.4.1.412.100.3.1.2.137 NAME
'dlm1AdjacentSlotsHelperAuxClass'
 DESC 'Helper class for finding AdjacentSlots.'
 SUP top AUXILIARY
```

```
MAY (dlmAdjacentSlotsHelperRef)
)
```

## 4 References

Request For Comments (RFC) and Internet Draft documents are available from numerous mirror sites.

[1] CIM, "CIM Physical Model, v2.5,"

[http://www.dmtf.org/spec/CIM\\_Schema25/CIM\\_Physical25.mof](http://www.dmtf.org/spec/CIM_Schema25/CIM_Physical25.mof).

[2] M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)," RFC 2251, December 1997.

[3] Wahl, S. Kille, T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names," RFC 2253, December 1997

[4] DMTF, "DMTF LDAP Schema for the CIM v2.4 Core Information Model," <http://www.dmtf.org/spec/DEN/DSP0117.doc>, November 2000.

[5] M. Wahl, "A Summary of the X.500(96) User Schema for use with LDAPv3," RFC 2256, December 1997.

[6] DMTF, "LDAP Mapping Guidelines",  
<http://www.dmtf.org/spec/denh.html>

## 5 Structural Rules

To aid the reader in mapping what structure rules have been defined and referenced, the following table lists sections and documents where they have been defined.

| Structure Rule  | Name                                          | Naming Attribute | Section |
|-----------------|-----------------------------------------------|------------------|---------|
| <physical-sr-1> | dIm1PhysicalPackageInstanceStructureRule      | orderedCimKeys   | 3.1     |
| <physical-sr-2> | dIm1PhysicalFrameInstanceStructureRule        | orderedCimKeys   | 3.2     |
| <physical-sr-3> | dIm1RackInstanceStructureRule                 | orderedCimKeys   | 3.3     |
| <physical-sr-4> | dIm1ChassisInstanceStructureRule              | orderedCimKeys   | 3.4     |
| <physical-sr-5> | dIm1CardInstanceStructureRule                 | orderedCimKeys   | 3.5     |
| <physical-sr-6> | dIm1SystemBusCardInstanceStructureRule        | orderedCimKeys   | 3.6     |
| <physical-      | dIm1StorageMediaLocationInstanceStructureRule | orderedCimKeys   | 3.7     |

|                  |                                                 |                |       |
|------------------|-------------------------------------------------|----------------|-------|
| sr-7>            |                                                 |                |       |
| <physical-sr-8>  | dIm1MagazineInstanceStructureRule               | orderedCimKeys | 3.8   |
| <physical-sr-9>  | dIm1PhysicalComponentInstanceStructureRule      | orderedCimKeys | 3.9   |
| <physical-sr-10> | dIm1ChipInstanceStructureRule                   | orderedCimKeys | 3.10  |
| <physical-sr-11> | dIm1PhysicalMemoryInstanceStructureRule         | orderedCimKeys | 3.11  |
| <physical-sr-12> | dIm1PhysicalMediaInstanceStructureRule          | orderedCimKeys | 3.12  |
| <physical-sr-13> | dIm1PhysicalTapeInstanceStructureRule           | orderedCimKeys | 3.13  |
| <physical-sr-14> | dIm1PhysicalLinkInstanceStructureRule           | orderedCimKeys | 3.14  |
| <physical-sr-15> | dIm1PhysicalConnectorInstanceStructureRule      | orderedCimKeys | 3.15  |
| <physical-sr-16> | dIm1SlotInstanceStructureRule                   | orderedCimKeys | 3.16  |
| <physical-sr-17> | dIm1MediaPhysicalStatInfoInstanceStructureRule  | orderedCimKeys | 3.17  |
| <physical-sr-18> | dIm1LocationInstanceStructureRule               | orderedCimKeys | 3.18  |
| <physical-sr-19> | dIm1MemoryCapacityInstanceStructureRule         | orderedCimKeys | 3.20  |
| <physical-sr-20> | dIm1ConfigurationCapacityInstanceStructureRule  | orderedCimKeys | 3.21  |
| <physical-sr-21> | dIm1ReplacementSetInstanceStructureRule         | orderedCimKeys | 3.22  |
| <physical-sr-22> | dIm1DeviceServicesLocationInstanceStructureRule | orderedCimKeys | 3.30  |
| <physical-sr-23> | dIm1ComputerSystemPackageInstanceStructureRule  | orderedCimKeys | 3.40  |
| <physical-sr-24> | dIm1AdjacentSlotsInstanceStructureRule          | orderedCimKeys | 3.57  |
| <physical-sr-25> | dImServicePhilosophyInstanceStructureRule       | ArrayIndex     | 2.3.1 |
| <physical-sr-26> | dImChassisTypeInstanceStructureRule             | arrayIndex     | 2.3.2 |
| <physical-sr-27> | dImMediaTypesSupportedInstanceStructureRule     | arrayIndex     | 2.3.3 |
| <physical-sr-28> | dImPhysicalLabelsInstanceStructureRule          | arrayIndex     | 2.3.4 |

## 6 OID Assignments

The following three tables provides the summary of OID assignments made in this document.

## 6.1 Object Classes

| <b>OID</b>                    | <b>Object Class Name</b>         | <b>Section</b> |
|-------------------------------|----------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.1.2.138 | dImServicePhilosophyInstance     | 2.3.1          |
| 1.3.6.1.4.1.412.100.3.1.2.139 | dImChassisTypeInstance           | 2.3.2          |
| 1.3.6.1.4.1.412.100.3.1.2.140 | dImMediaTypesSupportedInstance   | 2.3.3          |
| 1.3.6.1.4.1.412.100.3.1.2.141 | dImPhysicalLabelsInstance        | 2.3.4          |
| 1.3.6.1.4.1.412.100.3.1.2.1   | dIm1PhysicalPackage              | 3.1            |
| 1.3.6.1.4.1.412.100.3.1.2.2   | dIm1PhysicalPackageAuxClass      | 3.1            |
| 1.3.6.1.4.1.412.100.3.1.2.3   | dIm1PhysicalPackageInstance      | 3.1            |
| 1.3.6.1.4.1.412.100.3.1.2.4   | dIm1PhysicalFrame                | 3.2            |
| 1.3.6.1.4.1.412.100.3.1.2.5   | dIm1PhysicalFrameAuxClass        | 3.2            |
| 1.3.6.1.4.1.412.100.3.1.2.6   | dIm1PhysicalFrameInstance        | 3.2            |
| 1.3.6.1.4.1.412.100.3.1.2.7   | dIm1Rack                         | 3.3            |
| 1.3.6.1.4.1.412.100.3.1.2.8   | dIm1RackAuxClass                 | 3.3            |
| 1.3.6.1.4.1.412.100.3.1.2.9   | dIm1RackInstance                 | 3.3            |
| 1.3.6.1.4.1.412.100.3.1.2.10  | dIm1Chassis                      | 3.4            |
| 1.3.6.1.4.1.412.100.3.1.2.11  | dIm1ChassisAuxClass              | 3.4            |
| 1.3.6.1.4.1.412.100.3.1.2.12  | dIm1ChassisInstance              | 3.4            |
| 1.3.6.1.4.1.412.100.3.1.2.13  | dIm1Card                         | 3.5            |
| 1.3.6.1.4.1.412.100.3.1.2.14  | dIm1CardAuxClass                 | 3.5            |
| 1.3.6.1.4.1.412.100.3.1.2.15  | dIm1CardInstance                 | 3.5            |
| 1.3.6.1.4.1.412.100.3.1.2.16  | dIm1SystemBusCard                | 3.6            |
| 1.3.6.1.4.1.412.100.3.1.2.17  | dIm1SystemBusCardAuxClass        | 3.6            |
| 1.3.6.1.4.1.412.100.3.1.2.18  | dIm1SystemBusCardInstance        | 3.6            |
| 1.3.6.1.4.1.412.100.3.1.2.19  | dIm1StorageMediaLocation         | 3.7            |
| 1.3.6.1.4.1.412.100.3.1.2.20  | dIm1StorageMediaLocationAuxClass | 3.7            |
| 1.3.6.1.4.1.412.100.3.1.2.21  | dIm1StorageMediaLocationInstance | 3.7            |
| 1.3.6.1.4.1.412.100.3.1.2.22  | dIm1Magazine                     | 3.8            |
| 1.3.6.1.4.1.412.100.3.1.2.23  | dIm1MagazineAuxClass             | 3.8            |
| 1.3.6.1.4.1.412.100.3.1.2.24  | dIm1MagazineInstance             | 3.8            |
| 1.3.6.1.4.1.412.100.3.1.2.25  | dIm1PhysicalComponent            | 3.9            |
| 1.3.6.1.4.1.412.100.3.1.2.26  | dIm1PhysicalComponentAuxClass    | 3.9            |
| 1.3.6.1.4.1.412.100.3.1.2.27  | dIm1PhysicalComponentInstance    | 3.9            |
| 1.3.6.1.4.1.412.100.3.1.2.28  | dIm1Chip                         | 3.10           |
| 1.3.6.1.4.1.412.100.3.1.2.29  | dIm1ChipAuxClass                 | 3.10           |
| 1.3.6.1.4.1.412.100.3.1.2.30  | dIm1ChipInstance                 | 3.10           |
| 1.3.6.1.4.1.412.100.3.1.2.31  | dIm1PhysicalMemory               | 3.11           |
| 1.3.6.1.4.1.412.100.3.1.2.32  | dIm1PhysicalMemoryAuxClass       | 3.11           |
| 1.3.6.1.4.1.412.100.3.1.2.33  | dIm1PhysicalMemoryInstance       | 3.11           |

| <b>OID</b>                   | <b>Object Class Name</b>             | <b>Section</b> |
|------------------------------|--------------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.1.2.34 | d1m1PhysicalMedia                    | 3.12           |
| 1.3.6.1.4.1.412.100.3.1.2.35 | d1m1PhysicalMediaAuxClass            | 3.12           |
| 1.3.6.1.4.1.412.100.3.1.2.36 | d1m1PhysicalMediaInstance            | 3.12           |
| 1.3.6.1.4.1.412.100.3.1.2.37 | d1m1PhysicalTape                     | 3.13           |
| 1.3.6.1.4.1.412.100.3.1.2.38 | d1m1PhysicalTapeAuxClass             | 3.13           |
| 1.3.6.1.4.1.412.100.3.1.2.39 | d1m1PhysicalTapeInstance             | 3.13           |
| 1.3.6.1.4.1.412.100.3.1.2.40 | d1m1PhysicalLink                     | 3.14           |
| 1.3.6.1.4.1.412.100.3.1.2.41 | d1m1PhysicalLinkAuxClass             | 3.14           |
| 1.3.6.1.4.1.412.100.3.1.2.42 | d1m1PhysicalLinkInstance             | 3.14           |
| 1.3.6.1.4.1.412.100.3.1.2.43 | d1m1PhysicalConnector                | 3.15           |
| 1.3.6.1.4.1.412.100.3.1.2.44 | d1m1PhysicalConnectorAuxClass        | 3.15           |
| 1.3.6.1.4.1.412.100.3.1.2.45 | d1m1PhysicalConnectorInstance        | 3.15           |
| 1.3.6.1.4.1.412.100.3.1.2.46 | d1m1Slot                             | 3.16           |
| 1.3.6.1.4.1.412.100.3.1.2.47 | d1m1SlotAuxClass                     | 3.16           |
| 1.3.6.1.4.1.412.100.3.1.2.48 | d1m1SlotInstance                     | 3.16           |
| 1.3.6.1.4.1.412.100.3.1.2.49 | d1m1MediaPhysicalStatInfo            | 3.17           |
| 1.3.6.1.4.1.412.100.3.1.2.50 | d1m1MediaPhysicalStatInfoAuxClass    | 3.17           |
| 1.3.6.1.4.1.412.100.3.1.2.51 | d1m1MediaPhysicalStatInfoInstance    | 3.17           |
| 1.3.6.1.4.1.412.100.3.1.2.52 | d1m1Location                         | 3.18           |
| 1.3.6.1.4.1.412.100.3.1.2.53 | d1m1LocationAuxClass                 | 3.18           |
| 1.3.6.1.4.1.412.100.3.1.2.54 | d1m1LocationInstance                 | 3.18           |
| 1.3.6.1.4.1.412.100.3.1.2.55 | d1m1PhysicalCapacity                 | 3.19           |
| 1.3.6.1.4.1.412.100.3.1.2.56 | d1m1MemoryCapacity                   | 3.20           |
| 1.3.6.1.4.1.412.100.3.1.2.57 | d1m1MemoryCapacityAuxClass           | 3.20           |
| 1.3.6.1.4.1.412.100.3.1.2.58 | d1m1MemoryCapacityInstance           | 3.20           |
| 1.3.6.1.4.1.412.100.3.1.2.59 | d1m1ConfigurationCapacity            | 3.21           |
| 1.3.6.1.4.1.412.100.3.1.2.60 | d1m1ConfigurationCapacityAuxClass    | 3.21           |
| 1.3.6.1.4.1.412.100.3.1.2.61 | d1m1ConfigurationCapacityInstance    | 3.21           |
| 1.3.6.1.4.1.412.100.3.1.2.62 | d1m1ReplacementSet                   | 3.22           |
| 1.3.6.1.4.1.412.100.3.1.2.63 | d1m1ReplacementSetAuxClass           | 3.22           |
| 1.3.6.1.4.1.412.100.3.1.2.64 | d1m1ReplacementSetInstance           | 3.22           |
| 1.3.6.1.4.1.412.100.3.1.2.65 | d1m1RealizesExtent                   | 3.23           |
| 1.3.6.1.4.1.412.100.3.1.2.66 | d1m1RealizesExtentAuxClass           | 3.23           |
| 1.3.6.1.4.1.412.100.3.1.2.67 | d1m1RealizesPExtent                  | 3.24           |
| 1.3.6.1.4.1.412.100.3.1.2.68 | d1m1RealizesPExtentAuxClass          | 3.24           |
| 1.3.6.1.4.1.412.100.3.1.2.69 | d1m1RealizesDiskPartition            | 3.25           |
| 1.3.6.1.4.1.412.100.3.1.2.70 | d1m1RealizesDiskPartitionAuxClass    | 3.25           |
| 1.3.6.1.4.1.412.100.3.1.2.71 | d1m1RealizesAggregatePExtent         | 3.26           |
| 1.3.6.1.4.1.412.100.3.1.2.72 | d1m1RealizesAggregatePExtentAuxClass | 3.26           |
| 1.3.6.1.4.1.412.100.3.1.2.73 | d1m1RealizesTapePartition            | 3.27           |
| 1.3.6.1.4.1.412.100.3.1.2.74 | d1m1RealizesTapePartitionAuxClass    | 3.27           |



| <b>OID</b>                    | <b>Object Class Name</b>                 | <b>Section</b> |
|-------------------------------|------------------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.1.2.75  | dIm1RealizedOnSide                       | 3.28           |
| 1.3.6.1.4.1.412.100.3.1.2.76  | dIm1RealizedOnSideAuxClass               | 3.28           |
| 1.3.6.1.4.1.412.100.3.1.2.77  | dIm1Docked                               | 3.29           |
| 1.3.6.1.4.1.412.100.3.1.2.78  | dIm1DockedAuxClass                       | 3.29           |
| 1.3.6.1.4.1.412.100.3.1.2.79  | dIm1DeviceServicesLocation               | 3.30           |
| 1.3.6.1.4.1.412.100.3.1.2.80  | dIm1DeviceServicesLocationInstance       | 3.30           |
| 1.3.6.1.4.1.412.100.3.1.2.81  | dIm1DeviceServicesLocationHelperAuxClass | 3.30           |
| 1.3.6.1.4.1.412.100.3.1.2.82  | dIm1MemoryWithMedia                      | 3.31           |
| 1.3.6.1.4.1.412.100.3.1.2.83  | dIm1MemoryWithMediaAuxClass              | 3.31           |
| 1.3.6.1.4.1.412.100.3.1.2.84  | dIm1PhysicalMediaInLocation              | 3.32           |
| 1.3.6.1.4.1.412.100.3.1.2.85  | dIm1PhysicalMediaInLocationAuxClass      | 3.32           |
| 1.3.6.1.4.1.412.100.3.1.2.86  | dIm1ElementsLinked                       | 3.33           |
| 1.3.6.1.4.1.412.100.3.1.2.87  | dIm1ElementsLinkedAuxClass               | 3.33           |
| 1.3.6.1.4.1.412.100.3.1.2.88  | dIm1ConnectedTo                          | 3.34           |
| 1.3.6.1.4.1.412.100.3.1.2.89  | dIm1ConnectedToAuxClass                  | 3.34           |
| 1.3.6.1.4.1.412.100.3.1.2.90  | dIm1SlotInSlot                           | 3.35           |
| 1.3.6.1.4.1.412.100.3.1.2.91  | dIm1SlotInSlotAuxClass                   | 3.35           |
| 1.3.6.1.4.1.412.100.3.1.2.92  | dIm1PackageInConnector                   | 3.36           |
| 1.3.6.1.4.1.412.100.3.1.2.93  | dIm1PackageInConnectorAuxClass           | 3.36           |
| 1.3.6.1.4.1.412.100.3.1.2.94  | dIm1PackageInSlot                        | 3.37           |
| 1.3.6.1.4.1.412.100.3.1.2.95  | dIm1PackageInSlotAuxClass                | 3.37           |
| 1.3.6.1.4.1.412.100.3.1.2.96  | dIm1CardInSlot                           | 3.38           |
| 1.3.6.1.4.1.412.100.3.1.2.97  | dIm1CardInSlotAuxClass                   | 3.38           |
| 1.3.6.1.4.1.412.100.3.1.2.98  | dIm1AdapterActiveConnection              | 3.39           |
| 1.3.6.1.4.1.412.100.3.1.2.99  | dIm1AdapterActiveConnectionAuxClass      | 3.39           |
| 1.3.6.1.4.1.412.100.3.1.2.100 | dIm1ComputerSystemPackage                | 3.40           |
| 1.3.6.1.4.1.412.100.3.1.2.101 | dIm1ComputerSystemPackageInstance        | 3.40           |
| 1.3.6.1.4.1.412.100.3.1.2.102 | dIm1ComputerSystemPackageHelperAuxClass  | 3.40           |
| 1.3.6.1.4.1.412.100.3.1.2.103 | dIm1LibraryPackage                       | 3.41           |
| 1.3.6.1.4.1.412.100.3.1.2.104 | dIm1LibraryPackageAuxClass               | 3.41           |
| 1.3.6.1.4.1.412.100.3.1.2.105 | dIm1PackageCooling                       | 3.42           |
| 1.3.6.1.4.1.412.100.3.1.2.106 | dIm1PackageCoolingAuxClass               | 3.42           |
| 1.3.6.1.4.1.412.100.3.1.2.107 | dIm1PackageTempSensor                    | 3.43           |
| 1.3.6.1.4.1.412.100.3.1.2.108 | dIm1PackageTempSensorAuxClass            | 3.43           |
| 1.3.6.1.4.1.412.100.3.1.2.109 | dIm1PackageAlarm                         | 3.44           |
| 1.3.6.1.4.1.412.100.3.1.2.110 | dIm1PackageAlarmAuxClass                 | 3.44           |
| 1.3.6.1.4.1.412.100.3.1.2.111 | dIm1HomeForMedia                         | 3.45           |
| 1.3.6.1.4.1.412.100.3.1.2.112 | dIm1HomeForMediaAuxClass                 | 3.45           |
| 1.3.6.1.4.1.412.100.3.1.2.113 | dIm1Container                            | 3.46           |
| 1.3.6.1.4.1.412.100.3.1.2.114 | dIm1ContainerAuxClass                    | 3.46           |
| 1.3.6.1.4.1.412.100.3.1.2.115 | dIm1ChassisInRack                        | 3.47           |

| <b>OID</b>                    | <b>Object Class Name</b>            | <b>Section</b> |
|-------------------------------|-------------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.1.2.116 | d1m1ChassisInRackAuxClass           | 3.47           |
| 1.3.6.1.4.1.412.100.3.1.2.117 | d1m1PackageInChassis                | 3.48           |
| 1.3.6.1.4.1.412.100.3.1.2.118 | d1m1PackageInChassisAuxClass        | 3.48           |
| 1.3.6.1.4.1.412.100.3.1.2.119 | d1m1CardOnCard                      | 3.49           |
| 1.3.6.1.4.1.412.100.3.1.2.120 | d1m1CardOnCardAuxClass              | 3.49           |
| 1.3.6.1.4.1.412.100.3.1.2.121 | d1m1PackagedComponent               | 3.50           |
| 1.3.6.1.4.1.412.100.3.1.2.122 | d1m1PackagedComponentAuxClass       | 3.50           |
| 1.3.6.1.4.1.412.100.3.1.2.123 | d1m1MemoryOnCard                    | 3.51           |
| 1.3.6.1.4.1.412.100.3.1.2.124 | d1m1MemoryOnCardAuxClass            | 3.51           |
| 1.3.6.1.4.1.412.100.3.1.2.125 | d1m1ConnectorOnPackage              | 3.52           |
| 1.3.6.1.4.1.412.100.3.1.2.126 | d1m1ConnectorOnPackageAuxClass      | 3.52           |
| 1.3.6.1.4.1.412.100.3.1.2.127 | d1m1LinkHasConnector                | 3.53           |
| 1.3.6.1.4.1.412.100.3.1.2.128 | d1m1LinkHasConnectorAuxClass        | 3.53           |
| 1.3.6.1.4.1.412.100.3.1.2.129 | d1m1PhysicalElementLocation         | 3.54           |
| 1.3.6.1.4.1.412.100.3.1.2.130 | d1m1PhysicalElementLocationAuxClass | 3.54           |
| 1.3.6.1.4.1.412.100.3.1.2.131 | d1m1ElementCapacity                 | 3.55           |
| 1.3.6.1.4.1.412.100.3.1.2.132 | d1m1ElementCapacityAuxClass         | 3.55           |
| 1.3.6.1.4.1.412.100.3.1.2.133 | d1m1ParticipatesInSet               | 3.56           |
| 1.3.6.1.4.1.412.100.3.1.2.134 | d1m1ParticipatesInSetAuxClass       | 3.56           |
| 1.3.6.1.4.1.412.100.3.1.2.135 | d1m1AdjacentSlots                   | 3.57           |
| 1.3.6.1.4.1.412.100.3.1.2.136 | d1m1AdjacentSlotsInstance           | 3.57           |
| 1.3.6.1.4.1.412.100.3.1.2.137 | d1m1AdjacentSlotsHelperAuxClass     | 3.57           |

## 6.2 Attributes

| <b>OID</b>                   | <b>Attribute Name</b>  | <b>Section</b> |
|------------------------------|------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.2.1015 | d1mServicePhilosophy   | 2.3.1          |
| 1.3.6.1.4.1.412.100.3.2.1014 | d1mServiceDescriptions | 2.3.1          |
| 1.3.6.1.4.1.412.100.3.2.1019 | d1mChassisTypes        | 2.3.2          |
| 1.3.6.1.4.1.412.100.3.2.1023 | d1mTypeDescriptions    | 2.3.2          |
| 1.3.6.1.4.1.412.100.3.2.1035 | d1mMediaSizesSupported | 2.3.3          |
| 1.3.6.1.4.1.412.100.3.2.1036 | d1mMediaTypesSupported | 2.3.3          |
| 1.3.6.1.4.1.412.100.3.2.1037 | d1mLabelFormats        | 2.3.4          |
| 1.3.6.1.4.1.412.100.3.2.1038 | d1mLabelStates         | 2.3.4          |
| 1.3.6.1.4.1.412.100.3.2.1039 | d1mPhysicalLabels      | 2.3.4          |
| 1.3.6.1.4.1.412.100.3.2.1001 | d1mDepth               | 3.1            |
| 1.3.6.1.4.1.412.100.3.2.1002 | d1mHeight              | 3.1            |
| 1.3.6.1.4.1.412.100.3.2.1003 | d1mHotSwappable        | 3.1            |
| 1.3.6.1.4.1.412.100.3.2.1004 | d1mRemovable           | 3.1            |
| 1.3.6.1.4.1.412.100.3.2.1005 | d1mReplaceable         | 3.1            |
| 1.3.6.1.4.1.412.100.3.2.1006 | d1mWeight              | 3.1            |
| 1.3.6.1.4.1.412.100.3.2.1007 | d1mWidth               | 3.1            |

| <b>OID</b>                   | <b>Attribute Name</b>         | <b>Section</b> |
|------------------------------|-------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.2.1008 | dImAudibleAlarm               | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1009 | dImBreachDescription          | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1010 | dImCableManagementStrategy    | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1011 | dImIsLocked                   | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1012 | dImLockPresent                | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1013 | dImSecurityBreach             | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1016 | dImVisibleAlarm               | 3.2            |
| 1.3.6.1.4.1.412.100.3.2.1018 | dImTypeOfRack                 | 3.3            |
| 1.3.6.1.4.1.412.100.3.2.1020 | dImCurrentRequiredOrProduced' | 3.4            |
| 1.3.6.1.4.1.412.100.3.2.1021 | dImHeatGeneration             | 3.4            |
| 1.3.6.1.4.1.412.100.3.2.1022 | dImNumberOfPowerCords         | 3.4            |
| 1.3.6.1.4.1.412.100.3.2.1024 | dImHostingBoard               | 3.5            |
| 1.3.6.1.4.1.412.100.3.2.1025 | dImOperatingVoltages          | 3.5            |
| 1.3.6.1.4.1.412.100.3.2.1026 | dImRequirementsDescription    | 3.5            |
| 1.3.6.1.4.1.412.100.3.2.1027 | dImRequiresDaughterBoard      | 3.5            |
| 1.3.6.1.4.1.412.100.3.2.1028 | dImSlotLayout                 | 3.5            |
| 1.3.6.1.4.1.412.100.3.2.1029 | dImSpecialRequirements        | 3.5            |
| 1.3.6.1.4.1.412.100.3.2.1030 | dImBusType                    | 3.6            |
| 1.3.6.1.4.1.412.100.3.2.1031 | dImBusWidth                   | 3.6            |
| 1.3.6.1.4.1.412.100.3.2.1032 | dImLocationCoordinates        | 3.7            |
| 1.3.6.1.4.1.412.100.3.2.1033 | dImLocationType               | 3.7            |
| 1.3.6.1.4.1.412.100.3.2.1034 | dImMediaCapacity              | 3.7            |
| 1.3.6.1.4.1.412.100.3.2.1040 | dImFormFactor                 | 3.10           |
| 1.3.6.1.4.1.412.100.3.2.1041 | dImBankLabel                  | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1042 | dImCapacity                   | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1043 | dImDataWidth                  | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1044 | dImInterleavePosition         | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1045 | dImMemoryType                 | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1046 | dImPositionInRow              | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1047 | dImSpeed                      | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1048 | dImTotalWidth                 | 3.11           |
| 1.3.6.1.4.1.412.100.3.2.1049 | dImCleanerMedia               | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1050 | dImDualSided                  | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1051 | dImMaxMounts                  | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1052 | dImMediaDescription           | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1053 | dImMediaSize                  | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1054 | dImMediaType                  | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1055 | dImMountCount                 | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1056 | dImTimeOfLastMount            | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1057 | dImTotalMountTime             | 3.12           |
| 1.3.6.1.4.1.412.100.3.2.1058 | dImWriteProtectOn             | 3.12           |

| <b>OID</b>                   | <b>Attribute Name</b>           | <b>Section</b> |
|------------------------------|---------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.2.1059 | dImTapeLength                   | 3.13           |
| 1.3.6.1.4.1.412.100.3.2.1060 | dImUnloadAnywhere               | 3.13           |
| 1.3.6.1.4.1.412.100.3.2.1061 | dImLength                       | 3.14           |
| 1.3.6.1.4.1.412.100.3.2.1062 | dImMaxLength                    | 3.14           |
| 1.3.6.1.4.1.412.100.3.2.1063 | dImWired                        | 3.14           |
| 1.3.6.1.4.1.412.100.3.2.1064 | dImConnectorPinout              | 3.15           |
| 1.3.6.1.4.1.412.100.3.2.1065 | dImConnectorType                | 3.15           |
| 1.3.6.1.4.1.412.100.3.2.1066 | dImOtherTypeDescription         | 3.15           |
| 1.3.6.1.4.1.412.100.3.2.1067 | dImHeightAllowed                | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1068 | dImLengthAllowed                | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1069 | dImMaxDataWidth                 | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1070 | dImNumber                       | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1071 | dImOpenSwitch                   | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1072 | dImPowered                      | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1073 | dImPurposeDescription           | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1074 | dImSpecialPurpose               | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1075 | dImSupportsHotPlug              | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1076 | dImThermalRating                | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1077 | dImVccMixedVoltageSupport       | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1078 | dImVppMixedVoltageSupport       | 3.16           |
| 1.3.6.1.4.1.412.100.3.2.1079 | dImPickFailures                 | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1080 | dImPickRetries                  | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1081 | dImPickSuccesses                | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1082 | dImPutFailures                  | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1083 | dImPutRetries                   | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1084 | dImPutSuccesses                 | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1085 | dImScanFailures                 | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1086 | dImScanRetries                  | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1087 | dImScanSuccesses                | 3.17           |
| 1.3.6.1.4.1.412.100.3.2.1088 | dImAddress                      | 3.18           |
| 1.3.6.1.4.1.412.100.3.2.1089 | dImPhysicalPosition             | 3.18           |
| 1.3.6.1.4.1.412.100.3.2.1091 | dImMaximumMemoryCapacity        | 3.20           |
| 1.3.6.1.4.1.412.100.3.2.1092 | dImMinimumMemoryCapacity        | 3.20           |
| 1.3.6.1.4.1.412.100.3.2.1093 | dImIncrement                    | 3.21           |
| 1.3.6.1.4.1.412.100.3.2.1094 | dImMaximumCapacity              | 3.21           |
| 1.3.6.1.4.1.412.100.3.2.1095 | dImMinimumCapacity              | 3.21           |
| 1.3.6.1.4.1.412.100.3.2.1096 | dImObjectType                   | 3.21           |
| 1.3.6.1.4.1.412.100.3.2.1097 | dImStartingAddress              | 3.23           |
| 1.3.6.1.4.1.412.100.3.2.1098 | dImRealizesExtentAntecedentRef  | 3.23           |
| 1.3.6.1.4.1.412.100.3.2.1099 | dImRealizesExtentDependentRef   | 3.23           |
| 1.3.6.1.4.1.412.100.3.2.1100 | dImRealizesPExtentAntecedentRef | 3.24           |

| <b>OID</b>                   | <b>Attribute Name</b>                    | <b>Section</b> |
|------------------------------|------------------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.2.1101 | dImRealizesPExtentDependentRef           | 3.24           |
| 1.3.6.1.4.1.412.100.3.2.1102 | dImRealizesDiskPartitionAntecedentRef    | 3.25           |
| 1.3.6.1.4.1.412.100.3.2.1103 | dImRealizesDiskPartitionDependentRef     | 3.25           |
| 1.3.6.1.4.1.412.100.3.2.1104 | dImRealizesAggregatePExtentAntecedentRef | 3.26           |
| 1.3.6.1.4.1.412.100.3.2.1105 | dImRealizesAggregatePExtentDependentRef  | 3.26           |
| 1.3.6.1.4.1.412.100.3.2.1106 | dImRealizesTapePartitionAntecedentRef    | 3.27           |
| 1.3.6.1.4.1.412.100.3.2.1107 | dImRealizesTapePartitionDependentRef     | 3.27           |
| 1.3.6.1.4.1.412.100.3.2.1108 | dImSide                                  | 3.28           |
| 1.3.6.1.4.1.412.100.3.2.1109 | dImRealizedOnSideAntecedentRef           | 3.28           |
| 1.3.6.1.4.1.412.100.3.2.1110 | dImRealizedOnSideDependentRef            | 3.28           |
| 1.3.6.1.4.1.412.100.3.2.1111 | dImDockedAntecedentRef                   | 3.29           |
| 1.3.6.1.4.1.412.100.3.2.1112 | dImDockedDependentRef                    | 3.29           |
| 1.3.6.1.4.1.412.100.3.2.1113 | dImInaccessible                          | 3.30           |
| 1.3.6.1.4.1.412.100.3.2.1114 | dImDeviceServicesLocationAntecedentRef   | 3.30           |
| 1.3.6.1.4.1.412.100.3.2.1115 | dImDeviceServicesLocationDependentRef    | 3.30           |
| 1.3.6.1.4.1.412.100.3.2.1116 | dImDeviceServicesLocationHelperRef       | 3.30           |
| 1.3.6.1.4.1.412.100.3.2.1117 | dImMemoryWithMediaAntecedentRef          | 3.31           |
| 1.3.6.1.4.1.412.100.3.2.1118 | dImMemoryWithMediaDependentRef           | 3.31           |
| 1.3.6.1.4.1.412.100.3.2.1119 | dImOrientation                           | 3.32           |
| 1.3.6.1.4.1.412.100.3.2.1120 | dImTimeOfLastPlacement                   | 3.32           |
| 1.3.6.1.4.1.412.100.3.2.1121 | dImPhysicalMediaInLocationAntecedentRef  | 3.32           |
| 1.3.6.1.4.1.412.100.3.2.1122 | dImPhysicalMediaInLocationDependentRef   | 3.32           |
| 1.3.6.1.4.1.412.100.3.2.1123 | dImElementsLinkedAntecedentRef           | 3.33           |
| 1.3.6.1.4.1.412.100.3.2.1124 | dImElementsLinkedDependentRef            | 3.33           |
| 1.3.6.1.4.1.412.100.3.2.1125 | dImConnectedToAntecedentRef              | 3.34           |
| 1.3.6.1.4.1.412.100.3.2.1126 | dImConnectedToDependentRef               | 3.34           |
| 1.3.6.1.4.1.412.100.3.2.1127 | dImSlotInSlotAntecedentRef               | 3.35           |
| 1.3.6.1.4.1.412.100.3.2.1128 | dImSlotInSlotDependentRef                | 3.35           |
| 1.3.6.1.4.1.412.100.3.2.1129 | dImPackageInConnectorAntecedentRef       | 3.36           |
| 1.3.6.1.4.1.412.100.3.2.1130 | dImPackageInConnectorDependentRef        | 3.36           |
| 1.3.6.1.4.1.412.100.3.2.1131 | dImPackageInSlotAntecedentRef            | 3.37           |
| 1.3.6.1.4.1.412.100.3.2.1132 | dImPackageInSlotDependentRef             | 3.37           |
| 1.3.6.1.4.1.412.100.3.2.1133 | dImCardInSlotAntecedentRef               | 3.38           |
| 1.3.6.1.4.1.412.100.3.2.1134 | dImCardInSlotDependentRef                | 3.38           |
| 1.3.6.1.4.1.412.100.3.2.1135 | dImAdapterActiveConnectionAntecedentRef  | 3.39           |
| 1.3.6.1.4.1.412.100.3.2.1136 | dImAdapterActiveConnectionDependentRef   | 3.39           |
| 1.3.6.1.4.1.412.100.3.2.1137 | dImPlatformGUID                          | 3.40           |
| 1.3.6.1.4.1.412.100.3.2.1138 | dImComputerSystemPackageAntecedentRef    | 3.40           |
| 1.3.6.1.4.1.412.100.3.2.1139 | dImComputerSystemPackageDependentRef     | 3.40           |
| 1.3.6.1.4.1.412.100.3.2.1140 | dImComputerSystemPackageHelperRef        | 3.40           |
| 1.3.6.1.4.1.412.100.3.2.1141 | dImLibraryPackageAntecedentRef           | 3.41           |

| <b>OID</b>                   | <b>Attribute Name</b>                         | <b>Section</b> |
|------------------------------|-----------------------------------------------|----------------|
| 1.3.6.1.4.1.412.100.3.2.1142 | dImLibraryPackageDependentRef                 | 3.41           |
| 1.3.6.1.4.1.412.100.3.2.1143 | dImPackageCoolingAntecedentRef                | 3.42           |
| 1.3.6.1.4.1.412.100.3.2.1144 | dImPackageCoolingDependentRef                 | 3.42           |
| 1.3.6.1.4.1.412.100.3.2.1145 | dImPackageTempSensorAntecedentRef             | 3.43           |
| 1.3.6.1.4.1.412.100.3.2.1146 | dImPackageTempSensorDependentRef              | 3.43           |
| 1.3.6.1.4.1.412.100.3.2.1147 | dImPackageAlarmAntecedentRef                  | 3.44           |
| 1.3.6.1.4.1.412.100.3.2.1148 | dImPackageAlarmDependentRef                   | 3.44           |
| 1.3.6.1.4.1.412.100.3.2.1149 | dImHomeForMediaAntecedentRef                  | 3.45           |
| 1.3.6.1.4.1.412.100.3.2.1150 | dImHomeForMediaDependentRef                   | 3.45           |
| 1.3.6.1.4.1.412.100.3.2.1151 | dImLocationWithinContainer                    | 3.46           |
| 1.3.6.1.4.1.412.100.3.2.1152 | dImContainerGroupComponentRef                 | 3.46           |
| 1.3.6.1.4.1.412.100.3.2.1153 | dImContainerPartComponentRef                  | 3.46           |
| 1.3.6.1.4.1.412.100.3.2.1154 | dImBottomU                                    | 3.47           |
| 1.3.6.1.4.1.412.100.3.2.1155 | dImChassisInRackGroupComponentRef             | 3.47           |
| 1.3.6.1.4.1.412.100.3.2.1156 | dImChassisInRackPartComponentRef              | 3.47           |
| 1.3.6.1.4.1.412.100.3.2.1157 | dImPackageInChassisGroupComponentRef          | 3.48           |
| 1.3.6.1.4.1.412.100.3.2.1158 | dImPackageInChassisPartComponentRef           | 3.48           |
| 1.3.6.1.4.1.412.100.3.2.1159 | dImMountOrSlotDescription                     | 3.49           |
| 1.3.6.1.4.1.412.100.3.2.1160 | dImCardOnCardGroupComponentRef                | 3.49           |
| 1.3.6.1.4.1.412.100.3.2.1161 | dImCardOnCardPartComponentRef                 | 3.49           |
| 1.3.6.1.4.1.412.100.3.2.1162 | dImPackagedComponentGroupComponentRef         | 3.50           |
| 1.3.6.1.4.1.412.100.3.2.1163 | dImPackagedComponentPartComponentRef          | 3.50           |
| 1.3.6.1.4.1.412.100.3.2.1164 | dImMemoryOnCardGroupComponentRef              | 3.51           |
| 1.3.6.1.4.1.412.100.3.2.1165 | dImMemoryOnCardPartComponentRef               | 3.51           |
| 1.3.6.1.4.1.412.100.3.2.1166 | dImConnectorOnPackageGroupComponentRef        | 3.52           |
| 1.3.6.1.4.1.412.100.3.2.1167 | dImConnectorOnPackagePartComponentRef         | 3.52           |
| 1.3.6.1.4.1.412.100.3.2.1168 | dImLinkHasConnectorGroupComponentRef          | 3.53           |
| 1.3.6.1.4.1.412.100.3.2.1169 | dImLinkHasConnectorPartComponentRef           | 3.53           |
| 1.3.6.1.4.1.412.100.3.2.1170 | dImPhysicalElementLocationElementRef          | 3.54           |
| 1.3.6.1.4.1.412.100.3.2.1171 | dImPhysicalElementLocationPhysicalLocationRef | 3.54           |
| 1.3.6.1.4.1.412.100.3.2.1172 | dImElementCapacityCapacityRef                 | 3.55           |
| 1.3.6.1.4.1.412.100.3.2.1173 | dImElementCapacityElementRef                  | 3.55           |
| 1.3.6.1.4.1.412.100.3.2.1174 | dImParticipatesInSetElementRef                | 3.56           |
| 1.3.6.1.4.1.412.100.3.2.1175 | dImParticipatesInSetSetRef                    | 3.56           |
| 1.3.6.1.4.1.412.100.3.2.1176 | dImDistanceBetweenSlots                       | 3.57           |
| 1.3.6.1.4.1.412.100.3.2.1177 | dImSharedSlots                                | 3.57           |
| 1.3.6.1.4.1.412.100.3.2.1178 | dImAdjacentSlotsSlotARef                      | 3.57           |
| 1.3.6.1.4.1.412.100.3.2.1179 | dImAdjacentSlotsSlotBRef                      | 3.57           |
| 1.3.6.1.4.1.412.100.3.2.1180 | dImAdjacentSlotsHelperRef                     | 3.57           |

## 6.3 Naming Forms

| <b>OID</b>                 | <b>Name Form</b>                           | <b>Section</b> |
|----------------------------|--------------------------------------------|----------------|
| 1.3.6.1.4.412.100.3.3.2.25 | dImServicePhilosophyInstanceNameForm       | 2.3.1          |
| 1.3.6.1.4.412.100.3.3.2.26 | dImChassisTypeInstanceNameForm             | 2.3.2          |
| 1.3.6.1.4.412.100.3.3.2.27 | dImMediaTypesSupportedInstanceNameForm     | 2.3.3          |
| 1.3.6.1.4.412.100.3.3.2.28 | dImPhysicalLabelsInstanceNameForm          | 2.3.4          |
| 1.3.6.1.4.412.100.3.3.2.1  | dIm1PhysicalPackageInstanceNameForm        | 3.1            |
| 1.3.6.1.4.412.100.3.3.2.2  | dIm1PhysicalFrameInstanceNameForm          | 3.2            |
| 1.3.6.1.4.412.100.3.3.2.3  | dIm1RackInstanceNameForm                   | 3.3            |
| 1.3.6.1.4.412.100.3.3.2.4  | dIm1ChassisInstanceNameForm                | 3.4            |
| 1.3.6.1.4.412.100.3.3.2.5  | dIm1CardInstanceNameForm                   | 3.5            |
| 1.3.6.1.4.412.100.3.3.2.6  | dIm1SystemBusCardInstanceNameForm          | 3.6            |
| 1.3.6.1.4.412.100.3.3.2.7  | dIm1StorageMediaLocationInstanceNameForm   | 3.7            |
| 1.3.6.1.4.412.100.3.3.2.8  | dIm1MagazineInstanceNameForm               | 3.8            |
| 1.3.6.1.4.412.100.3.3.2.9  | dIm1PhysicalComponentInstanceNameForm      | 3.9            |
| 1.3.6.1.4.412.100.3.3.2.10 | dIm1ChipInstanceNameForm                   | 3.10           |
| 1.3.6.1.4.412.100.3.3.2.11 | dIm1PhysicalMemoryInstanceNameForm         | 3.11           |
| 1.3.6.1.4.412.100.3.3.2.12 | dIm1PhysicalMediaInstanceNameForm          | 3.12           |
| 1.3.6.1.4.412.100.3.3.2.13 | dIm1PhysicalTapeInstanceNameForm           | 3.13           |
| 1.3.6.1.4.412.100.3.3.2.14 | dIm1PhysicalLinkInstanceNameForm           | 3.14           |
| 1.3.6.1.4.412.100.3.3.2.15 | dIm1PhysicalConnectorInstanceNameForm      | 3.15           |
| 1.3.6.1.4.412.100.3.3.2.16 | dIm1SlotInstanceNameForm                   | 3.16           |
| 1.3.6.1.4.412.100.3.3.2.17 | DIm1MediaPhysicalStatInfoInstanceNameForm  | 3.17           |
| 1.3.6.1.4.412.100.3.3.2.18 | dIm1LocationInstanceNameForm               | 3.18           |
| 1.3.6.1.4.412.100.3.3.2.19 | dIm1MemoryCapacityInstanceNameForm         | 3.20           |
| 1.3.6.1.4.412.100.3.3.2.20 | dIm1ConfigurationCapacityInstanceNameForm  | 3.21           |
| 1.3.6.1.4.412.100.3.3.2.21 | dIm1ReplacementSetInstanceNameForm         | 3.22           |
| 1.3.6.1.4.412.100.3.3.2.22 | dIm1DeviceServicesLocationInstanceNameForm | 3.30           |
| 1.3.6.1.4.412.100.3.3.2.23 | dIm1ComputerSystemPackageInstanceNameForm  | 3.40           |
| 1.3.6.1.4.412.100.3.3.2.24 | dIm1AdjacentSlotsInstanceNameForm          | 3.57           |