Technical Study

Mobile and Directory Architecture

THE Open GROUP

Technical Study

Mobile and Directory Architecture

Document Number: E042

Published by The Open Group, April 2004.

Comments relating to the material contained in this document may be submitted to:

The Open Group
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by electronic mail to:

ogspecs@opengroup.org

Contents

Preface

The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX certification.

Further information on The Open Group is available at www.opengroup.org.

The Open Group has over 15 years' experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification.

More information is available at www.opengroup.org/testing.

The Open Group publishes a wide range of technical documentation, the main part of which is focused on development of Technical and Product Standards and Guides, but which also includes white papers, technical studies, branding and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/pubs.

As with all *live* documents, Technical Standards and Specifications require revision to align with new developments and associated international standards. To distinguish between revised specifications which are fully backwards-compatible and those which are not:

- A new *Version* indicates there is no change to the definitive information contained in the previous publication of that title, but additions/extensions are included. As such, it *replaces* the previous publication.

- A new *Issue* indicates there is substantive change to the definitive information contained in the previous publication of that title, and there may also be additions/extensions. As such, both previous and new documents are maintained as current publications.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

This Document

This Technical Study presents an architecture for use by service providers and enterprises to support mobile computing.

The architecture is based on the use of directory technology to make information about users and the network available to applications, management systems, and intelligent network components. It includes the detailed definition of a directory schema appropriate for this purpose.

The schema has been partly validated by being loaded into a number of different directory products, but the architecture as a whole has yet to be fully deployed in a practical situation.

This Technical Study:

- Reviews the requirements
- Outlines the development of the architecture
- Describes the architecture
- Discusses identity management issues
- Describes the directory schema

The detailed schema definition is given in Appendix A.

Mobile telephony has become a commonplace of modern life. Mobile computing is less well established, partly because of the lack of a common standard infrastructure. The architecture described in this Technical Study could be the basis of a common standard infrastructure that answers this need.

Trademarks

Boundaryless Information Flow™ is a trademark and UNIX® and The Open Group® are registered trademarks of The Open Group in the United States and other countries.

The Open Group acknowledges that there may be other brand, company, and product names used in this document that may be covered by trademark protection and advises the reader to verify them independently.

1 Introduction

Mobile telephony has become a commonplace of modern life. Mobile computing is less well established, partly because of the lack of a common standard infrastructure. This Technical Study describes an architecture that could be the basis of a common standard infrastructure for mobile computing, for deployment by service providers and enterprises.

It grew from work in The Open Group Mobile Management Forum (MMF). The MMF developed a demonstration of seamless session management across different networks, which was given at The Open Group Conference in Berlin, April 2001.

This led to the development by the MMF together with The Open Group Directory Interoperability Forum (DIF) of the Business Scenario – *The Executive on the Move* – that described the requirements for directories to support mobile computing.

The Open Group then worked on a Challenge to vendors of directories and related products to provide support for mobile computing. The idea of the Challenge received broad support. The Distributed Management Task Force (DMTF) and the Network Applications Consortium (NAC) joined The Open Group as Challenge partners.

The architecture described in this Technical Study was developed by the Challenge team. Its central feature is the use of directory technology to make information about users and the network available to applications, management systems, and intelligent network components. A directory schema appropriate for this purpose has been developed and worked out in detail.

The architecture has been validated, in part, by the successful loading of the directory schema into a number of different directories. But the team has not yet produced a public demonstration of the architecture as a full and final response to the Challenge.

This Technical Study is published to provide a description of the architecture that can be studied and referenced by parties outside the Challenge team. It:

- Reviews the requirements
- Outlines the development of the architecture
- Describes the architecture
- Discusses identity management issues
- Describes the directory schema

The detailed schema definition is given in Appendix A.

2 The Requirements

Mobile computing provides many benefits to business users when they are away from their normal places of work. Fixed and wireless Internet connection facilities can be found today in many hotels and airports, and at conferences. Some enterprises allow visitors to connect through their networks. The traveler finds it increasingly easy to remain in contact through email and via the web.

The communications infrastructure that supports this is operated by enterprises and by service providers. Directories can form a useful part of that infrastructure.

This section reviews:

- Why business users need mobile computing
- What the requirements are on the supporting infrastructure
- What directories must do if they are to be part of it

It incorporates the conclusions of the Business Scenario – *The Executive on the Move* – that was developed based on a wide range of inputs from vendor and user organizations. That Business Scenario includes more background and rationale than there is space for here.

2.1 User Requirements

This architecture is intended to improve support for Internet access by people away from their normal places of work.

Increasingly, mobile computing and communications are used by executives and other employees of a wide range of government, commercial, manufacturing, service, and other organizations. They help to:

- Minimize time when people are not doing productive, profitable work
- Maximize availability of executives for decision-making
- Maximize competitive advantage by timely availability of information
- Improve quality of enterprise information

A traveling user will move from one service provider to another. For example, someone traveling to a meeting may connect from home in the morning, from an airport lounge while traveling, from the meeting in the afternoon, and from a hotel in the evening.

Users have requirements for communications functionality, security, ubiquity, convenience, and quality of service. These requirements vary from one user to another; each user has an individual requirements profile.

2.2 Service Provider Requirements

Service providers – including enterprises as well as specialist Internet service providers – must meet the users' requirements. They must also meet regulatory requirements, which vary from one country to another. They need to operate and manage their networks. Finally, specialist service providers want to charge customers for the services that they use.

Today, service providers are able to provide simple Internet connectivity. Where they charge for it, they do so by a variety of means, and usually independently of each other. They cannot deliver individually profiled services, and the user pays separately and individually for access through each provider. This is inconvenient and expensive when the user passes through the areas of several providers in a short time.

To deliver and manage individually profiled services, providers must make decisions based on a combination of information about the state of the network and the identity of the user.

To deliver consolidated charging, they must share information about users.

Directories can help enterprises and specialist service providers to do these things.

2.3 Directory Requirements

The requirements on directories that are used to support mobile computing are that they should hold information about users and the network available to applications and network components in a consistent and well-structured manner.

Putting the information that supports mobility into a directory means that:

- Applications and intelligent network components have ready access to the information through a well-defined interface
- Administrators can configure the information together with related information that may be stored in directories, particularly information about people

Storing information about users and their use of the network in a consistent and well-structured manner facilitates the sharing of that information for consolidated billing.

3 Development of the Architecture

The Business Scenario – *The Executive on the Move* –identified the need for a standard directory schema. It also identified a major difficulty. Directories are not very well suited to store information – such as network state – that changes rapidly in real time.

A standard directory schema was proposed by the Distributed Management Task Force (DMTF). It is derived from their Common Information Model (CIM). It provides for the standardized representation in directories of the network information needed to support mobile computing. It was adopted, and partly validated, by the Challenge team.

The problem of storing in directories information that changes rapidly was the subject of much thought. The Business Scenario proposed use of a back-end data store designed to hold rapidly changing information, and accessible via the directory. A later idea, better because it involved more use of components available off-the-shelf, was to use a virtual directory front end, and to access the directory and the network information through it.

A Network Operations Center (NOC) would collect network information and the virtual directory would obtain it from the NOC. The advantage of this, as opposed to having the virtual directory interface to the network components directly, is that the virtual directory just needs a single plug-in, which enables it to interface indirectly to network components of all kinds, rather than needing a proprietary plug-in for each make and kind of network component.

This single plug-in uses Web-Based Enterprise Management (WBEM) and should be relatively straightforward to implement. It translates between the Challenge subset of the CIM schema and the Lightweight Directory Access Protocol (LDAP) mapping of this subset. It probably does not need to support the whole of WBEM, just instance retrieval query.

Using the NOC to relay the information should not introduce a serious performance overhead.

Two further design issues were raised: the desire to support seamless session handover, and the need to provide a standard identity management framework for service providers to enable authentication and consolidated billing.

Seamless session handover between service providers is a desirable requirement for the long term, but users can probably live without it for the present. It is a difficult problem, and one whose solution will not largely depend on directories, although directories may play a part. The Secure Mobile Architecture (SMA) proposed by The Open Group Mobile Management Forum (MMF) provides a possible solution. The Challenge team decided not to incorporate seamless session handover in the Challenge initially, but to leave open the possibility of incorporating it at a later stage if and when others develop a solution. This Technical Study does not address it.

The need to provide a standard identity management framework is partly addressed by the standard schema described in this Technical Study. This, however, only addresses storage of the information, not communication of the information between service providers. Also, it may not address all of the information storage requirements. Taking the solution further would require a better understanding of those requirements, which can only be obtained from service providers.

This architecture therefore does not address seamless session handover, and only partly addresses identity management. It does, however, address the storage and access by applications and network components of information about the network and its users.

4 Architecture Description

Figure 1 shows how the architecture would be instantiated by a service provider. When demonstrated in the Challenge, it would include an additional component – the Challenge Display (also shown) – to enable the audience to follow the demonstration.

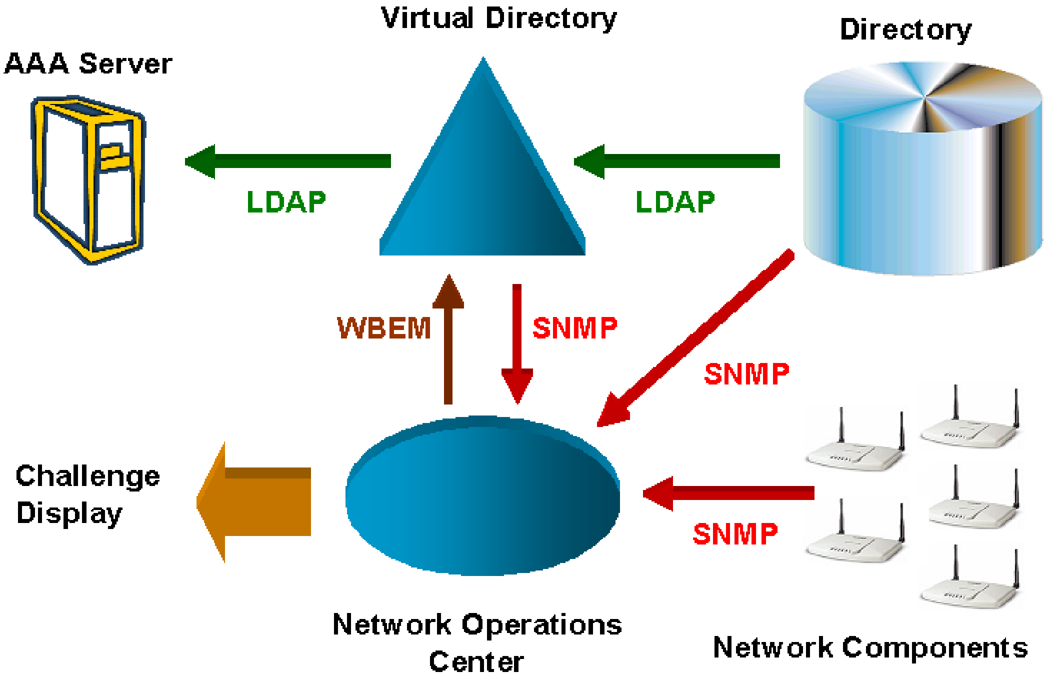

Figure 1: Challenge Service Provider Architecture

The "AAA Server" tells the access point network components what traffic to allow, making policy decisions based on information retrieved from the directory. The precise mechanism for this requires further discussion, and may depend on the access points used. A likely mechanism[1] is to configure the access points to allow only traffic to and from devices with specified Media Access Control (MAC) addresses. Alternatively, or in addition, Extensible Authentication Protocol (EAP) authentication could be employed.

The access points communicate with the AAA server using the Remote Authentication Dial-In User Service (RADIUS). The allowable users' MAC addresses and/or EAP shared secrets are held in their directory entries, and retrieved by the AAA server using the Lightweight Directory Access Protocol (LDAP).

The AAA server can also gather accounting information.

[1] This mechanism is supported by the Cisco AP 340 Access Points that were offered for use in the Challenge.

The “Directory” holds information about users and may also hold information about the network configuration.

Network configuration information stored in the directory consists of static information only. It could include what components are installed, how they are connected, and their physical locations.

Note that this information could be stored in the Network Operations Center (NOC), rather than in the directory.

Network components such as wireless access points hold information about themselves for management purposes. The directory and virtual directory also hold such information. This information can be accessed via the Simple Network Management Protocol (SNMP).

The formats of the information for various different kinds of device are defined in Management Information Base (MIB) standards.

The NOC is a CIM Object Manager (CIMOM), as defined by the Distributed Management Task Force (DMTF). It obtains information from the network components, and also from the directory and virtual directory, using SNMP.

The NOC handles SNMP and translates the information defined by the MIB standards to the form described by the Challenge schema. (Future generations of network component may support the Common Information Model [CIM] directly. Currently available network components just support SNMP, so that a translation is necessary.)

The changing network and user information is displayed by the NOC to the audience during the Challenge, so that they can see what is happening.

The “Virtual Directory” provides information to support authentication, access control, and accounting to the AAA server using LDAP.

In a real-world deployment of this architecture, it could also provide information about users and the network to applications and intelligent network components using LDAP.

It obtains information about users and static network information from the directory using LDAP.

It obtains the rapidly changing network information from the NOC using the CIM operations over Hyper Text Transfer Protocol (HTTP) defined by the DMTF for Web-Based Enterprise Management (WBEM). If static network information is held in the NOC rather than the directory, then the virtual directory obtains it from the NOC in a similar way.

The virtual-directory/directory combination might be replaced by a metadirectory or by a metadirectory/directory combination.

5 Identity Management Issues

Users may authenticate themselves to the networks on certain occasions, such as when registering with a new network. But it is impractical to authenticate users separately for each network transaction and, indeed, the network components are typically unaware of the identities of the users whose traffic they are handling. Network components will be aware either of the users' equipment identities (e.g., Media Access Control [MAC] addresses), or of their IP addresses, or perhaps of both.

IP addresses are assigned to users when they start using access networks – via the Dynamic Host Configuration Protocol (DHCP). When a user moves from one access network to another, a new IP address is allocated. More precisely, the IP addresses are assigned not to the users but to their communication endpoints. In practice, these endpoints will probably be wireless networking cards in PCs.

The typical scenario today is that a user authenticates to a service provider, or assures the service provider of payment. (A service provider may not need to know who the user is, provided that payment is assured.) The service provider allocates an IP address to the user's PC network interface, and gives the user access to the network.

This process is not particularly secure. There are various ways in which a user's IP address can be usurped, or his traffic interfered with. But it is good enough for everyday operation. (The process of using tickets to authorize rail transportation is not particularly secure. There are various ways in which tickets can be stolen, or checks on them can be evaded. But it is good enough for everyday operation.)

As a matter of convenience, it is desirable to avoid the user having to re-authenticate when moving from one service provider to another. For the purpose of the Challenge demonstration, it was proposed to allow the user's equipment identity (MAC address) to serve as identification of the user for authentication purposes. A possible addition to this, enabling more secure operation, is authentication by the Extensible Authentication Protocol (EAP). A user can then authenticate to one service provider and move to another without re-authenticating. The second service provider will note that a new piece of equipment has joined its network and obtain from the first service provider details of the user associated with the equipment identifier.

This is not particularly secure, but is good enough for demonstration purposes (and perhaps even for everyday use also).

More secure means of identifying users and their equipment are possible, and there are industry initiatives investigating them. For example:

- The Host Identity Payload (HIP) provides a mechanism by which the identity of the equipment or its user is carried securely in every packet. The Open Group Secure Mobile Architecture (SMA) incorporates this mechanism.

- EAP can be used in conjunction with other authentication mechanisms to establish an identity that can have global significance. The EAP Subscriber Identity Module (SIM) work in the IETF describes how EAP can be used in conjunction with mobile telephony SIM algorithms to authenticate mobile subscriber identities.

The architecture described here is believed to be generally compatible with such initiatives. They are, however, beyond the scope of the present Technical Study.

The association between user identity, equipment identity, EAP shared secret (if used), and IP address is stored in the service providers' directories. This is one of the advantages of a directory-based architecture.

Transfer of information between service providers can, for demonstration purposes, take place through a simple Lightweight Directory Access Protocol (LDAP) query. It may be that for deployment in real life a form of federation between the service providers' directories, with information exchange by a protocol such as SAML, as proposed by the Liberty Alliance, will be more appropriate. This is for further investigation.

6 The Schema

6.1 Overview

The Challenge schema forms a crucial part of the architecture. It is derived from the Distributed Management Task Force (DMTF) Common Information Model (CIM).

The CIM is an object-oriented information model, published by the DMTF. Its goal is to describe all the various aspects of a managed environment – both computing and networking concepts. The model can be translated into specific formats appropriate for different repositories (such as directories, or transmission over the Hyper Text Transfer Protocol [HTTP], encoded in XML).

The schema of this architecture is based on a subset of CIM Version 2.7; that subset dictated by the information requirements of the Challenge scenario. In addition, several new classes have been added to the object hierarchy, reflecting aspects of the architecture that are specific to the Challenge or that are still "experimental" in CIM today.

The classes of the schema describe the following aspects of the managed environment:

- Core: Basic abstractions and meta-data
- Physical: Physical locations, packaging (for example, cards and chassis), and the relationships between them
- Logical: Software assets/inventory, general computer systems, services (such as the Dynamic Host Configuration Protocol [DHCP]), and the Ethernet, IP, and Wireless Endpoints in a system
- Interoperability/Management Infrastructure: Data provided by the Network Operations Center (NOC) and minimal information about the directories in the Challenge
- Users, Organizations, and Security: Credential and principal/identity information, including associations assigning identity to a "real" person
- Policy: General if-then policy rules (where the "if" clause is expressed using subclasses of *PolicyConditions*, and the "then" clause by *PolicyActions*)

The information in the schema is defined using CIM classes directly in the NOC implementation (i.e., a CIM Object Manager), but is also defined as a directory schema for the virtual directory. The directory schema is structured using the Directory-Enabled Networks (DEN) LDAP Mapping guidelines, documented in the DMTF DEN Technical Note.

It is straightforward to add new attributes and classes to a directory. So, it is understood how to add new aspects of the schema. However, there are some directory concepts that are not "new". For example, most directories already include object classes for people (typically *inetOrgPerson*), with attributes for role and security information. In these cases, the auxiliary form of a class in the architecture schema is applied to the corresponding directory schema classes and instances. The result is that new architecture schema attributes and relationships are attached through the auxiliary class, but the existing directory structure is maintained.

A similar mechanism is used for other kinds of information for which directory object classes are already defined.

The architecture schema has been partly validated, by being successfully loaded into a wide range of directories. This shows the consistency of the schema. It does not show that the schema represents the correct set of data needed to support mobile computing. This would be shown by the Challenge demonstration.

Also, the allocation of schema elements between the virtual directory and the directory has not been defined. This would need to be done prior to a practical instantiation of the architecture.

6.2 Definition

The model for the schema is defined in Appendix A. This definition uses an ASCII language called Managed Object Format (MOF), which is defined in the CIM specification by the DMTF. MOF is a human- and machine-readable description of an object-oriented diagram.

The architecture schema is obtained by translating this model into a language that can be understood by a directory. There is a directory language that can be used for schema definitions: the LDAP Data Interchange Format (LDIF). Although a standard version is defined by the IETF, most directory products implement slightly different variants of it. The translation must therefore be different for each directory product in which the schema is loaded.

Files containing ASCII versions of the model definition (in MOF) and of an algorithmically-derived schema definition that can be loaded into the open source OpenLDAP directory (in LDIF) are publicly available on The Open Group website.

The model definition comprises:

A.1	*MaD_CIMSchema11.mof*	A master file that calls up the others as include files
A.2	*Core27_Qualifiers.mof*	Meta-data qualifiers (meta-information for classes, properties, and methods)
A.3	*MaD_Core11.mof*	Core classes (subset of CIM Core Version 2.7)
A.4	*MaD_ManagementSAP11.mof*	Management service access point
A.5	*MaD_Physical11.mof*	Physical information
A.6	*MaD_Logical11.mof*	Logical information
A.7	*MaD_Interop11.mof*	Interoperability information
A.8	*MaD_User11.mof*	User information
A.9	*MaD_Policy11.mof*	Policy information

This model “pulls” the relevant classes from the CIM Version 2.7 Final Definition, “upgrades” several classes based on the CIM Version 2.8 Preliminary Schema, and extends via a few new subclasses. Each CIM class is named “CIM_...”, while the Challenge-specific extension classes are titled “MaD_...”. The Version information for each CIM class indicates the last CIM Release in which the definition of the class was updated. (“2.7.1000” indicates the CIM Version 2.8 Preliminary Release.)

The DMTF Directory Enabled Networks (DEN) project has defined a directory schema that is derived algorithmically from the whole of the CIM. The Challenge schema is separate from this DEN schema, but uses all the same “rules” of the conversion. In fact, there is a 1-1 correspondence between the part of the Challenge schema that is derived from CIM Version 2.7, and those same classes in the DEN schema. The corresponding classes in the two schemas differ by name (“xMAD...” *versus* “dlm...”) and object identifiers.

The Challenge schema files were processed separately from the DEN CIM Version 2.7 LDIFs. This is because the Challenge schema preceded the DEN CIM Version 2.7 schema, and used the “new” algorithms being put in place for the DEN Version 2.7 release.

Glossary and References

AAA Authentication, Authorization, and Accounting.

AP Access Point (for wireless communication).

ASCII American Standard Code for Information Interchange. An encoding commonly used to represent the letters of the Latin alphabet, numbers, punctuation symbols, etc. in computer systems. ISO/IEC 646[2] contains the standard definition.

Business Scenario
The Executive on the Move – see www.opengroup.org/publications/catalog/e042.htm.

CIM The DMTF Common Information Model. Information on the CIM can be found on the DMTF website at www.dmtf.org.

DEN Directory-Enabled Networks. The DMTF CIM in a directory environment is referred to as DEN. It is a CIM schema mapping, defining a directory schema – see www.dmtf.org/standards/den.

DHCP Dynamic Host Configuration Protocol defined by the IETF in RFC 2131 – see www.ietf.org/rfc/rfc2131.txt.

DIF The Open Group Directory Interoperability Forum – see www.opengroup.org/dif.

DMTF Distributed Management Task Force – see www.dmtf.org.

DMTF DEN Technical Note
The Technical Note at www.dmtf.org/education/technote_DENMapping.pdf.

EAP Extensible Authentication Protocol defined by the IETF in RFC 2284 – see www.ietf.org/rfc/rfc2284.txt.

EAP SIM Work-in-progress in the IETF on authentication and session key distribution using the mobile telephony Subscriber Identity Module.

EMA Electronic Messaging Association. This organization is now The Open Group Messaging Forum – see www.opengroup.org/messaging.

HTTP Hyper Text Transfer Protocol of the World Wide Web – see www.w3.org/Protocols.

IEEE Institute of Electrical and Electronics Engineers, Inc. – see www.ieee.org.

[2] ISO/IEC 646:1991, Information Processing – ISO 7-bit Coded Character Set for Information Interchange.

IETF
Internet Engineering Task Force – see www.ietf.org.

LAN
Local Area Network.

LDAP
Lightweight Directory Access Protocol defined by the IETF in RFC 2251 – see www.ietf.org/rfc/rfc2251.txt.

LDIF
LDAP Data Interchange Format. The standard version is defined by the IETF in RFC 2849 – see www.ietf.org/rfc/rfc2849.txt.

MAC
Media Access Control. A layer of the IEEE 802 series LAN protocols.[3]

MIB
Management Information Base. The MIB concept is defined by the IETF in RFC 1156 – see www.ietf.org/rfc/rfc1156.txt. RFCs specifying particular MIBs have since appeared and can be found on the IETF website at www.ietf.org/rfc.html.

MMF
The Open Group Mobile Management Forum – see www.opengroup.org/mobile.

Model Definition
ASCII files containing the model definition in MOF form are available on The Open Group website at www.opengroup.org/dif/docs/madmof01.zip.

MOF
Managed Object Format. A human- and machine-readable description of an object-oriented diagram, defined in the CIM Specification by the DMTF.

NAC
The Network Applications Consortium – see www.netapps.org.

NOC
Network Operations Center.

Open Pegasus
A CIM/WBEM Manageability Services Broker open source project hosted by The Open Group – see www.openpegasus.org.

PC
Personal Computer.

RADIUS
Remote Authentication Dial-In User Service defined by the IETF in RFC 2865 – see www.ietf.org/rfc/rfc2865.txt.

RFC
Request For Comments. Name used for specifications and other documents produced by the IETF.

Schema Definition
ASCII files containing a schema definition that can be loaded into the open source *OpenLDAP* directory in LDIF form are available on The Open Group website at www.opengroup.org/dif/docs/madldf01.zip.

SMA
Secure Mobile Architecture developed by the MMF – see www.opengroup.org/products/publications/catalog/e041.htm.

[3] IEEE Standards can be purchased from the IEEE – see standards.ieee.org.

SNMP Simple Network Management Protocol defined by the IETF in RFC 1157 – see www.ietf.org/rfc/rfc1157.txt.

The Open Group
See www.opengroup.org.

WBEM Web-Based Enterprise Management initiative of the DMTF. Information on WBEM can be found on the DMTF website at www.dmtf.org. The specification of CIM operations over HTTP for WBEM can be found at www.dmtf.org/standards/documents/WBEM/DSP200.html.

WLAN Wireless Local Area Network.

XML Extensible Markup Language. A generic information-representation format, derived from the Standard Generalized Markup Language, which is defined in ISO 8879.[4] Its specifications are published by the World-Wide Web Consortium – see www.w3.org/XML.

[4] ISO 8879:1986, Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML).

A Schema Model Formal Definition

A.1 Master File

```
// ===================================================================
// Title:       Mobile and Directory Challenge Schema
// Filename:    MaD_CIMSchema11.mof
// Version:     1.1.0
// Status:      Preliminary
// Date:        11/10/2003
// Description: MOF file to define necessary schema to support the
//              Mobile and Directory Challenge sponsored by The Open
//              Group, the Distributed Management Task Force, and
//              the Network Applications Consortium.
// Change History:
//   Feb 2003 - Original release
//   Nov 2003 - Move to CIM Version 2.7 Final Release classes;
//              Additions of ServiceAccessURI, ManagementSAP,
//              SecuritySensitivity and authentication policy;
//              Modifications to bring identity and security
//              classes in line with CIM Version 2.8 definitions.
// ===================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ===================================================================

// ===================================================================
// Includes
// ===================================================================

// Qualifiers defining class, property, and method "meta-data"
#pragma include ("Core27_Qualifiers.mof")

// Core classes as the "top of the tree" - Most are abstract
// ManagedElement, Dependency/ConcreteDependency,
//     Component/ConcreteComponent, LogicalIdentity/ConcreteIdentity,
//     Collection, MemberOfCollection, ManagedSystemElement,
//     LogicalElement, EnabledLogicalElement, System, AdminDomain,
//     SystemComponent, LogicalDevice, Realizes, SystemDevice,
//     Service, HostedService, ServiceComponent,
//     ServiceServiceDependency, ServiceAvailableToElement,
//     ServiceAccessPoint, ServiceAccessURI, RemoteServiceAccessPoint,
//     RemotePort, HostedAccessPoint, ServiceAccessBySAP,
//     ActiveConnection, ProtocolEndpoint, BindsTo, SettingData,
//     Profile, ElementSettingData, and ElementProfile
//     (from Core27_CoreElements.mof)
#pragma include ("MaD_Core11.mof")
```

```
// ManagementSAP (Can associate access points for manageability,
//     New in CIM Version 2.9)
#pragma include ("MaD_ManagementSAP11.mof")

// PhysicalElement, Location, ElementLocation, ContainedLocation,
//     and SystemPackaging (from Core27_Physical.mof)
// Several new properties in PhysicalElement (from Core28_Physical.mof)
// PhysicalPackage, Container, PhysicalFrame, Rack, Chassis
//     (from Physical27_Package.mof)
// Card (Physical27_Card.mof)
#pragma include ("MaD_Physical11.mof")

// Miscellaneous - Logical information
//     SoftwareIdentity (from Core27_Software.mof),
//     ComputerSystem (from System27_SystemElements.mof),
//     LANEndpoint, and IPProtocolEndpoint
//     (from Network27_ProtocolEndpoints.mof), and
//     LogicalPort, NetworkPort, and PortOnDevice classes
//     (from Device27_Ports.mof)
// New subclass of SettingData - UserChallengePreferences
// New classes for DHCP and wireless information - DHCPService,
//     DHCPAddress, DHCPServiceManagesAddress, AddressLease,
//     WirelessPort, and WirelessLANEndpoint
#pragma include ("MaD_Logical11.mof")

// WBEMService, ObjectManager, ObjectManagerCommunicationMechanism,
//     CIMXMLCommunicationMechanism, CommMechanismForManager,
//     Namespace, SystemIdentification, NamespaceInManager, and
//     SystemInNamespace (from CIM_Interop27.mof)
// New classes for Directory information - DirectoryService,
//     DirectoryCommunicationMechanism, and CommMechanismForDirectory
#pragma include ("MaD_Interop11.mof")

// OrganizationalEntity, Organization, OrgStructure, and UserEntity
//     (from User27_Org.mof)
// UserContact (from User28_Org.mof)
// Credential (from User27_Credential.mof)
// New classes for identity - Identity, SecuritySensitivity, and
//     AssignedIdentity (from CIM Version 2.8)
#pragma include ("MaD_User11.mof")

// New classes for authentication policy (from CIM Version 2.8)
#pragma include ("MaD_Policy11.mof")


// ==================================================================
// end of file
// ==================================================================
```

A.2 Metadata Qualifiers

```
// ==================================================================
// Title:       Core Qualifiers 2.7
// Filename:    Core27_Qualifiers.mof
// Version:     2.7.2
// Release:     Final
// Date:        05/01/03
// ==================================================================
// Copyright 1998-2003 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,
// provided that correct attribution is given.
// As DMTF specifications may be revised from time to time,
// the particular version and release date should always be noted.
//
// Implementation of certain elements of this standard or proposed
// standard may be subject to third-party patent rights, including
// provisional patent rights (herein "patent rights"). DMTF makes
// no representations to users of the standard as to the existence
// of such rights, and is not responsible to recognize, disclose, or
// identify any or all such third-party patent right, owners, or
// claimants, nor for any incomplete or inaccurate identification or
// disclosure of such rights, owners, or claimants. DMTF shall have no
// liability to any party, in any manner or circumstance, under any
// legal theory whatsoever, for failure to recognize, disclose, or
// identify any such third-party patent rights, or for such party's
// reliance on the standard or incorporation thereof in its product,
// protocols, or testing procedures. DMTF shall have no liability to
// any party implementing such standard, whether such implementation
// is foreseeable or not, nor to any patent owner or claimant, and
// shall have no liability or responsibility for costs or losses
// incurred if a standard is withdrawn or modified after publication,
// and shall be indemnified and held harmless by any party implementing
// the standard from any and all claims of infringement by a patent
// owner for such implementations.
//
// For information about patents held by third parties which have
// notified the DMTF that, in their opinion, such patent may relate to
// or impact implementations of DMTF standards, visit
// www.dmtf.org/about/policies/disclosures.php.
// ==================================================================
// Description: The Core Model defines basic management concepts.
//              This file defines the CIM qualifiers.
//
//              The object classes below are listed in an order that
//              avoids forward references. Required objects, defined
//              by other working groups, are omitted.
// =================================================================
// Prerequisite: None
// ==================================================================
// Change Log for Version 2.7
// CR664 - Change the default value for Version qualifier to NULL.
//       - Change the default value for Revision qualifier to NULL.
//       - Add a comment above the Revision qualifier stating that it
//         has been deprecated.
// CR707 - Add the Composition qualifier.
// CR713 - Clean up of the Scope and Flavors for Abstract, ArrayType,
//         EmbeddedObject, Experimental, and OctetString qualifiers.
// CR715 - Clean up of the Scope and Flavors for the Deprecated
//         Qualifier.
// CR716 - Add the MinLen qualifier.
// CR762 - Clean up of the Scope and Flavors for Association, DN,
//         OctetString, EmbeddedObject, Expensive, Indication,
//         MappingString, Modelcorrespondence, Required, and Terminal
//         qualifiers.
//
// Change Log for Version 2.7.2 - ERRATA
// CR1038 - Define the Experimental and Version qualifiers as
//          Restricted.
// ==================================================================

#pragma locale ("en_US")

// ==================================================================
// Qualifiers
// ==================================================================
Qualifier Abstract : boolean = false,
    Scope(class, association, indication),
    Flavor(Restricted);

Qualifier Aggregate: boolean = false,
    Scope (reference),
    Flavor(DisableOverride);

Qualifier Aggregation : boolean = false,
    Scope(association),
    Flavor(DisableOverride);

Qualifier Alias : string = null,
    Scope(property, reference, method),
    Flavor(Translatable);

Qualifier ArrayType : string = "Bag",
    Scope(property, parameter),
    Flavor(DisableOverride);

Qualifier Association : boolean = false,
    Scope(association),
    Flavor(DisableOverride);

Qualifier BitMap : string[],
    Scope(property, method, parameter);
```

```
Qualifier BitValues : string[],
    Scope(property, method, parameter),
    Flavor(Translatable);

Qualifier Composition : boolean = false,
    Scope(association),
    Flavor(DisableOverride);

Qualifier Counter : boolean = false,
    Scope(property, method, parameter);

Qualifier Delete : boolean = false,
    Scope(association, reference);

Qualifier Deprecated : string [],
    Scope(any),
    Flavor(Restricted);

Qualifier Description : string = null,
    Scope(any),
    Flavor(Translatable);

Qualifier DisplayName : string = null,
    Scope(any),
    Flavor(Translatable);

Qualifier DN : boolean=false,
    Scope(property, method, parameter),
    Flavor(DisableOverride);

Qualifier EmbeddedObject : boolean = false,
    Scope(property, method, parameter),
    Flavor(DisableOverride);

Qualifier Expensive : boolean = false,
    Scope(any);

Qualifier Experimental : boolean = false,
    Scope(any),
    Flavor(Restricted);

Qualifier Gauge : boolean = false,
    Scope(property, method, parameter);

Qualifier Ifdeleted : boolean = false,
    Scope(association, reference);

Qualifier In : boolean = true,
    Scope(parameter),
    Flavor(DisableOverride);
```

```
Qualifier Indication : boolean = false,
    Scope(class, indication),
    Flavor(DisableOverride);

Qualifier Invisible : boolean = false,
    Scope(reference, association, class, property, method);

Qualifier Key : boolean = false,
    Scope(property, reference),
    Flavor(DisableOverride);

Qualifier Large : boolean = false,
    Scope(property, class);

Qualifier MappingStrings : string[],
    Scope(any);

Qualifier Max : uint32 = null,
    Scope(reference);

Qualifier MaxLen : uint32 = null,
    Scope(property, method, parameter);

Qualifier MaxValue : sint64 = null,
    Scope(property, method, parameter);

Qualifier Min : uint32 = 0,
    Scope(reference);

Qualifier MinLen : uint32 = 0,
    Scope(property, method, parameter);

Qualifier MinValue : sint64 = null,
    Scope(property, method, parameter);

Qualifier ModelCorrespondence : string[],
    Scope(any);

Qualifier Nonlocal : string = null,
    Scope(reference);

Qualifier NonlocalType : string = null,
    Scope(reference);

Qualifier NullValue : string = null,
    Scope (property),
    Flavor(DisableOverride);

Qualifier Octetstring : boolean = false,
    Scope(property, method, parameter),
    Flavor(DisableOverride);
```

```
Qualifier Out : boolean = false,
    Scope(parameter),
    Flavor(DisableOverride);

Qualifier Override : string = null,
    Scope(property, method, reference),
    Flavor(DisableOverride);

Qualifier Propagated : string = null,
    Scope(property),
    Flavor(DisableOverride);

Qualifier PropertyUsage : string = "CurrentContext",
    Scope(property);

Qualifier Provider : string = null,
    Scope(any);

Qualifier Read : boolean = true,
    Scope(property);

Qualifier Required : boolean = false,
    Scope(property, reference, parameter),
    Flavor(DisableOverride);

// The Revision qualifier has been deprecated. For the replacement,
// see the usage rule for the version qualifier in the CIM
// Specification.
Qualifier Revision : string = null,
    Scope(class, association, indication ),
    Flavor(Translatable);

Qualifier Schema : string = null,
    Scope (property, method),
    Flavor(DisableOverride, Translatable);

Qualifier Source : string = null,
    Scope(class, association, indication);

Qualifier SourceType : string = null,
    Scope(class, association, indication, reference);

Qualifier Static : boolean = false,
    Scope(method, property),
    Flavor(DisableOverride);

Qualifier Syntax : string = null,
    Scope(property, reference, method, parameter);

Qualifier SyntaxType : string = null,
    Scope(property, reference, method, parameter);

Qualifier Terminal : boolean = FALSE,
    Scope(class, association, indication);
```

```
Qualifier TriggerType : string = null,
    Scope(class, property, reference, method, association, indication);

Qualifier Units : string = null,
    Scope(property, method, parameter),
    Flavor(Translatable);

Qualifier UnknownValues : string[],
    Scope(property),
    Flavor(DisableOverride);

Qualifier UnsupportedValues : string[],
    Scope(property),
    Flavor(DisableOverride);

Qualifier ValueMap : string[],
    Scope(property, method, parameter);

Qualifier Values : string[],
    Scope(property, method, parameter),
    Flavor(Translatable);

Qualifier Version : string = null,
    Scope(class, association, indication),
    Flavor(Restricted, Translatable);

Qualifier Weak : boolean = false,
    Scope(reference),
    Flavor(DisableOverride, ToSubclass);

Qualifier Write : boolean = false,
    Scope(property);

// ===================================================================
// end of file
// ===================================================================
```

A.3 Core Classes

```
// ==================================================================
// Title:       Mobile and Directory Schema - Subset of
//              CIM Core Version 2.7
// Filename:    MaD_Core11.mof
// Version:     1.1.0
// Release:     Preliminary
// Date:        11/10/03
// Change History:
//   Feb 2003 - Original release
//   Nov 2003 - Move to CIM Version 2.7 Final Release classes
//              including the addition of ServiceAccessURI.
// ==================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ==================================================================

#pragma locale ("en_US")

// ==================================================================
// ManagedElement
// ==================================================================
[Abstract, Version ("2.7.0"), Description (
    "ManagedElement is an abstract class that provides a common "
    "superclass (or top of the inheritance tree) for the "
    "non-association classes in the CIM Schema.") ]
class CIM_ManagedElement {

    [MaxLen (64), Description (
        "The Caption property is a short textual description (one-"
        "line string) of the object.") ]
    string Caption;

    [Description (
        "The Description property provides a textual description of "
        "the object.") ]
    string Description;

    [Description (
        "A user-friendly name for the object. This property allows "
        "each instance to define a user-friendly name IN ADDITION TO "
        "its key properties/identity data and description "
        "information. \n"
        "Note that ManagedSystemElement Name property is also defined "
        "as a user-friendly name. But, it is often subclassed to be a "
        "Key. It is not reasonable that the same property can convey "
        "both identity and a user friendly name without "
        "inconsistencies. "
        "Where Name exists and is not a Key (such as for instances of "
        "LogicalDevice), the same information MAY be present in both "
        "the Name and ElementName properties.") ]
    string ElementName;
};

// ==================================================================
// Dependency
// ==================================================================
[Association, Abstract, Version ("2.6.0"), Description (
    "CIM_Dependency is a generic association used to establish "
    "dependency relationships between ManagedElements.") ]
class CIM_Dependency {

    [Key, Description (
        "Antecedent represents the independent object in this "
        "association.") ]
    CIM_ManagedElement REF Antecedent;

    [Key, Description (
        "Dependent represents the object dependent on the "
        "Antecedent.") ]
    CIM_ManagedElement REF Dependent;
};

// ==================================================================
// ConcreteDependency
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "CIM_ConcreteDependency is a generic association used to "
    "establish dependency relationships between ManagedElements. It "
    "is defined as a concrete subclass of the abstract "
    "CIM_Dependency class, to be used in place of many specific "
    "subclasses of Dependency that add no semantics - "
    "i.e., that do not clarify the type of dependency, update "
    "cardinalities, or add/remove qualifiers. Note that when "
    "defining additional semantics for Dependency that this class "
    "MUST NOT be subclassed. Specific semantics continue to "
    "be defined as subclasses of the abstract CIM_Dependency. "
    "ConcreteDependency is limited in its use as a concrete form of "
    "a general dependency. \n"
    "\n"
    "It was deemed more prudent to create this concrete subclass "
    "than to change Dependency from an abstract to a concrete class. "
    "Dependency already had multiple abstract subclasses in the CIM "
    "Schema, and wider industry usage and impact could not be "
    "anticipated.") ]
class CIM_ConcreteDependency : CIM_Dependency {

    [Override ("Antecedent"), Description (
        "Antecedent represents the independent object in this "
        "association.") ]
    CIM_ManagedElement REF Antecedent;

    [Override ("Dependent"), Description (
        "Dependent represents the object dependent on the "
        "Antecedent.") ]
    CIM_ManagedElement REF Dependent;
};

// ==================================================================
// Component
// ==================================================================
[Association, Abstract, Aggregation, Version ("2.7.0"), Description (
    "CIM_Component is a generic association used to establish "
    "'part of' relationships between Managed Elements. For "
    "example, it could be used to define the components or parts of "
    "a System.") ]
class CIM_Component {

    [Aggregate, Key, Description (
        "The parent element in the association.") ]
    CIM_ManagedElement REF GroupComponent;

    [Key, Description (
        "The child element in the association.") ]
    CIM_ManagedElement REF PartComponent;
};

// ==================================================================
// ConcreteComponent
// ==================================================================
[Association, Aggregation,
    Version ("2.7.0"), Description (
    "CIM_ConcreteComponent is a generic association used to "
    "establish 'part of' relationships between ManagedElements. It "
    "is defined as a concrete subclass of the abstract "
    "CIM_Component class, to be used in place of many "
    "specific subclasses of Component that add no semantics - "
    "i.e., that do not clarify the type of composition, update "
    "cardinalities, or add/remove qualifiers. Note that when "
    "defining additional semantics for Component that this "
    "class MUST NOT be subclassed. Specific semantics "
    "continue to be defined as subclasses of the abstract "
    "CIM_Component. ConcreteComponent is limited in its use "
    "as a concrete form of a general composition. \n"
    "\n"
    "It was deemed more prudent to create this concrete subclass "
    "than to change Component from an abstract to a concrete class. "
    "Industry usage and impact could not be anticipated.") ]
class CIM_ConcreteComponent : CIM_Component {

    [Aggregate, Override ("GroupComponent"), Description (
        "The parent element in the association.") ]
    CIM_ManagedElement REF GroupComponent;

    [Override ("PartComponent"), Description (
        "The child element in the association.") ]
    CIM_ManagedElement REF PartComponent;
};
```

```
// ==================================================================
// LogicalIdentity
// ==================================================================
[Association, Abstract, Version ("2.7.0"), Description (
    "CIM_LogicalIdentity is an abstract and generic association, "
    "indicating that two ManagedElements represent different "
    "aspects of the same underlying entity. This relationship "
    "conveys what could be defined with multiple inheritance. "
    "In most scenarios, the Identity relationship is "
    "determined by the equivalence of Keys or some other "
    "identifying properties of the related Elements. \n\n"
    "One of the scenarios where this relationship is reasonable "
    "is to represent that a LogicalDevice is both a 'bus' entity and "
    "a 'functional' entity. For example, a Device could be both a "
    "USB (bus) and a Keyboard (functional) entity.") ]
class CIM_LogicalIdentity {

    [Key, Description (
        "SystemElement represents one aspect of the Managed"
        "Element. The use of 'System' in the role name does not "
        "limit the scope of the association. The role name was "
        "defined in the original association, where the "
        "referenced elements were limited to LogicalElements. "
        "Since that time, it has been found valuable to instantiate "
        "these types of relationships for ManagedElements, such as "
        "Collections. So, the referenced elements of the association "
        "were redefined to be ManagedElements. Unfortunately, the "
        "role name could not be changed without deprecating the "
        "entire association. This was not deemed necessary just to "
        "correct the role name.") ]
    CIM_ManagedElement REF SystemElement;

    [Key, Description (
        "SameElement represents an alternate aspect of the "
        "ManagedElement.") ]
    CIM_ManagedElement REF SameElement;
};

// ==================================================================
// ConcreteIdentity
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "CIM_ConcreteIdentity associates two elements representing "
    "different aspects of the same underlying entity. It is "
    "defined as a concrete subclass of the abstract "
    "CIM_LogicalIdentity class, to be used in place of many specific "
    "subclasses of LogicalIdentity that add no semantics; "
    "i.e., that do not clarify the type of identity, update "
    "cardinalities, or add/remove qualifiers. Note that when "
    "defining additional semantics for LogicalIdentity this class "
    "should not be subclassed. Specifc semantics continue to be "
    "defined as subclasses of the abstract CIM_LogicalIdentity. "
    "ConcreteIdentity is limited in its use as a concrete form of "
    "a general identity relationship. \n"
    "\n"
    "It was deemed more prudent to create this concrete subclass "
    "than to change LogicalIdentity from an abstract to a concrete "
    "class. LogicalIdentity already had multiple abstract subclasses "
    "in the CIM Schema, and wider industry usage and impact could "
    "not be anticipated.") ]
class CIM_ConcreteIdentity : CIM_LogicalIdentity {

    [Override ("SystemElement"), Description (
        "One aspect of the ManagedElement. The use of 'System' "
        "in the name does not limit the scope of the association. "
        "This is an artifact of the original definition of the "
        "association.") ]
    CIM_ManagedElement REF SystemElement;

    [Override ("SameElement"), Description (
        "Another aspect of the ManagedElement.") ]
    CIM_ManagedElement REF SameElement;
};

// ==================================================================
// ManagedSystemElement
// ==================================================================
[Abstract, Version ("2.7.0"), Description (
    "CIM_ManagedSystemElement is the base class for the System "
    "Element hierarchy. Membership Criteria: Any distinguishable "
    "component of a System is a candidate for inclusion in this "
    "class. Examples: software components, such as files; and "
    "devices, such as disk drives and controllers, and physical "
    "components such as chips and cards.") ]
class CIM_ManagedSystemElement : CIM_ManagedElement {

    [Description (
        "A datetime value indicating when the object was installed. "
        "A lack of a value does not indicate that the object is not "
        "installed."),
        MappingStrings {"MIF.DMTF|ComponentID|001.5"} ]
    datetime InstallDate;

    [MaxLen (1024), Description (
        "The Name property defines the label by which the object is "
        "known. When subclassed, the Name property can be overridden "
        "to be a Key property.") ]
    string Name;

    [Description (
        "Indicates the current status(es) of the element. "
        "Various health and operational statuses are "
        "defined. Many of the enumeration's values are self-"
        "explanatory. However, a few are not and are described "
        "in more detail. \n"
        " \"Stressed\" indicates that the element is functioning, "
        "but needs attention. Examples of \"Stressed\" states are "
        "overload, overheated, etc. \n"
        " \"Predictive Failure\" indicates that an element is "
        "functioning nominally but predicting a failure in the "
        "near future. \n"
        " \"In Service\" describes an element being configured, "
        "maintained, cleaned, or otherwise administered. \n"
        " \"No Contact\" indicates that the monitoring system "
        "has knowledge of this element, but has never been able to "
        "establish communications with it. \n"
        " \"Lost Communication\" indicates that the ManagedSystem"
        "Element is known to exist and has been contacted "
        "successfully in the past, but is currently unreachable. \n"
        " \"Stopped\" and \"Aborted\" are similar, although the "
        "former implies a clean and orderly stop, while the latter "
        "implies an abrupt stop where the element's state and "
        "configuration may need to be updated. \n"
        " \"Dormant\" indicates that the element is inactive or "
        "quiesced. \n"
        " \"Supporting Entity in Error\" describes that this "
        "element may be \"OK\" but that another element, on which it "
        "is dependent, is in error. An example is a network service "
        "or endpoint that cannot function due to lower layer "
        "networking problems. \n"
        " \"Completed\" indicates the element has completed its "
        "operation. This value should be combined with either OK, "
        "Error, or Degraded so that a client can tell if the complete "
        "operation passed (Completed with OK), and failure (Completed "
        "with Error). Completed with Degraded would imply the "
        "operation finished, but did not complete OK or report an "
        "error."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8",
            "9", "10", "11", "12", "13", "14", "15", "16", "17"},
        Values {"Unknown", "Other", "OK", "Degraded", "Stressed",
            "Predictive Failure", "Error", "Non-Recoverable Error",
            "Starting", "Stopping", "Stopped", "In Service",
            "No Contact", "Lost Communication", "Aborted",
            "Dormant", "Supporting Entity in Error", "Completed"},
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_ManagedSystemElement.StatusDescriptions"} ]
    uint16 OperationalStatus[];

    [Description (
        "Strings describing the various OperationalStatus "
        "array values. For example, if \"Stopping\" is the value "
        "assigned to OperationalStatus, then this property may "
        "contain an explanation as to why an object is being "
        "stopped. Note that entries in this array are correlated "
        "with those at the same array index in OperationalStatus."),
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_ManagedSystemElement.OperationalStatus"} ]
    string StatusDescriptions[];
};
```

```
// =====================================================================
// LogicalElement
// =====================================================================
[Abstract, Version ("2.6.0"), Description (
    "CIM_LogicalElement is a base class for all the components of "
    "a System that represent abstract system components, such "
    "as Files, Processes, or LogicalDevices.") ]
class CIM_LogicalElement : CIM_ManagedSystemElement {
};

// =====================================================================
// EnabledLogicalElement
// =====================================================================
[Abstract, Version ("2.7.0"), Description (
    "This class extends LogicalElement to abstract the concept of an "
    "element that is enabled and disabled, such as a LogicalDevice or "
    "a ServiceAccessPoint.") ]
class CIM_EnabledLogicalElement : CIM_LogicalElement {

    [Description (
        "EnabledState is an integer enumeration that indicates the "
        "enabled/disabled states of an element. It can also indicate "
        "the transitions between these requested states. For "
        "example, shutting down (value = 4) and starting "
        "(value=8) are transient states between enabled and disabled. "
        "The following text briefly summarizes the various "
        "enabled/disabled states: \n"
        "Enabled (2) indicates that the element is/could be executing "
        "commands, will process any queued commands, and queues new "
        "requests. \n"
        "Disabled (3) indicates that the element will not execute "
        "commands and will drop any new requests. \n"
        "Shutting Down (4) indicates that the element is in the "
        "process of going to a Disabled state. \n"
        "Not Applicable (5) indicates the element doesn't support "
        "being enabled/disabled. \n"
        "Enabled but Offline (6) indicates that the element may be "
        "completing commands, and will drop any new requests. \n"
        "Test (7) indicates that the element is in a test state. \n"
        "Deferred (8) indicates that the element may be completing "
        "commands, but will queue any new requests. \n"
        "Quiesce (9) indicates that the element is enabled but in a "
        "restricted mode. The element's behavior is similar to the "
        "Enabled state, but it only processes a restricted set of "
        "commands. All other requests are queued. \n"
        "Starting (10) indicates that the element is in the process "
        "of going to an Enabled state. New requests are queued. \n"),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10, 11..32767", "32768..65535" },
        Values {"Unknown", "Other", "Enabled", "Disabled",
            "Shutting Down", "Not Applicable", "Enabled but Offline",
            "In Test", "Deferred", "Quiesce", "Starting",
            "DMTF Reserved", "Vendor Reserved" },
        ModelCorrespondence {
            "CIM_EnabledLogicalElement.OtherEnabledState"} ]
    uint16 EnabledState = 5;

    [Description (
        "A string describing the element's enabled/disabled state "
        "when the EnabledState property is set to 1 (\"Other\"). "
        "This property MUST be set to NULL when EnabledState "
        "is any value other than 1."),
        ModelCorrespondence {"CIM_EnabledLogicalElement.EnabledState"}]
    string OtherEnabledState;

    [Description (
        "RequestedState is an integer enumeration indicating "
        "the last requested or desired state for the element. The "
        "actual state of the element is represented by EnabledState. "
        "This property is provided to compare the last requested and "
        "current enabled/disabled states. Note that when "
        "EnabledState is set to 5 (\"Not Applicable\"), "
        "then this property has no meaning. By default, "
        "the element's RequestedState is 5 (\"No Change\"). Refer "
        "to the EnabledState's property Description for explanations "
        "of the values in the RequestedState enumeration. \n"
        "\n"
        "This property is set as the result of a method invocation "
        "(such as Start or StopService on CIM_Service), or may be "
        "overridden and defined as WRITEable in a subclass. The "
        "method approach is considered superior to a WRITEable "
        "property, since it allows an explicit invocation of the "
        "operation and the return of a result code."),
        ModelCorrespondence {"CIM_EnabledLogicalElement.EnabledState"},
        ValueMap {"2", "3", "4", "5", "6", "7", "8", "9",
            "10..32767", "32768..65535" },
        Values {"Enabled", "Disabled", "Shut Down", "No Change",
            "Offline", "Test", "Deferred", "Quiesce",
            "DMTF Reserved", "Vendor Reserved"} ]
    uint16 RequestedState = 5;

    [Write, Description (
        "An enumerated value indicating an administrator's "
        "default/startup configuration for an element's Enabled"
        "State. By default, the element is \"Enabled\" (value=2)."),
        ValueMap {"2", "3", "5", "6", "7", "8..32767",
            "32768..65535"},
        Values {"Enabled", "Disabled", "Not Applicable",
            "Enabled but Offline", "No Default",
            "DMTF Reserved", "Vendor Reserved"} ]
    uint16 EnabledDefault = 2;
};

// ==================================================================
// Collection
// ==================================================================
[Abstract, Version ("2.6.0"), Description (
    "Collection is an abstract class that provides a common "
    "superclass for data elements that represent collections of "
    "ManagedElements and its subclasses.") ]
class CIM_Collection : CIM_ManagedElement {
};

// ==================================================================
// MemberOfCollection
// ==================================================================
[Association, Aggregation, Version ("2.6.0"), Description (
    "CIM_MemberOfCollection is an aggregation used to establish "
    "membership of ManagedElements in a Collection." ) ]
class CIM_MemberOfCollection {

    [Key, Aggregate, Description (
        "The Collection that aggregates members.") ]
    CIM_Collection REF Collection;

    [Key, Description (
        "The aggregated member of the Collection.") ]
    CIM_ManagedElement REF Member;
};

// ==================================================================
// System
// ==================================================================
[Abstract, Version ("2.7.0"), Description (
    "CIM_System represents an entity made up of component "
    "parts (defined by the SystemComponent relationship), that "
    "operates as a 'functional whole'. Systems are top level "
    "objects in the CIM hierarchy, requiring no scoping or weak "
    "relationships in order to exist and have context. It should "
    "be reasonable to uniquely name and manage a System at an "
    "enterprise level. For example, a ComputerSystem is a kind of "
    "System that can be uniquely named and independently managed "
    "in an enterprise. However, this is not true for the power "
    "supply (or the power supply sub-'system') within "
    "the computer. \n"
    "\n"
    "Although a System may be viewed as a Collection, this is "
    "not the correct model. A Collection is simply a 'bag' "
    "that 'holds' its members. A System is a higher level "
    "abstraction, built out of its individual components. It "
    "is more than a sum of its parts. Note that System is a "
    "subclass of EnabledLogicalElement which allows the entire "
    "abstraction to be functionally enabled/disabled - at a higher "
    "level than enabling/disabling its component parts.") ]
class CIM_System : CIM_EnabledLogicalElement {

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Key, MaxLen (256), Override ("Name"), Description (
        "The inherited Name serves as key of a System instance in "
        "an enterprise environment.") ]
    string Name;

    [MaxLen (64), Description (
        "The System object and its derivatives are Top Level Objects "
        "of CIM. They provide the scope for numerous components. "
        "Having unique System keys is required. A heuristic can be "
        "defined in individual System subclasses to attempt to always "
        "generate the same System Name Key. The NameFormat property "
        "identifies how the System name was generated, using "
        "the subclass' heuristic.") ]
    string NameFormat;

    [MaxLen (64), Write, Description (
        "The name of the primary system owner. The system owner "
        "is the primary user of the system."),
        MappingStrings {
            "MIF.DMTF|General Information|001.3"} ]
    string PrimaryOwnerName;

    [MaxLen (256), Write, Description (
        "A string that provides information on how the primary system "
        "owner can be reached (e.g., phone number, email address, "
        "etc.)."),
        MappingStrings {
            "MIF.DMTF|General Information|001.4"} ]
    string PrimaryOwnerContact;

    [Write, Description (
        "An array (bag) of strings that specify the administrator- "
        "defined roles this System plays in the managed environment. "
        "Examples might be 'Building 8 print server' or 'Boise user "
        "directories'. A single system may perform multiple roles. \n"
        "Note that instrumentation's view of a System's 'roles' is "
        "defined by instantiating a specific subclass of System and/"
        "or by properties in a subclass. For example, a "
        "ComputerSystem's purpose is defined using the Dedicated and "
        "OtherDedicatedDescription properties.") ]
    string Roles[];
};

// ==================================================================
// SystemComponent
// ==================================================================
[Association, Aggregation, Version ("2.7.0"), Description (
    "CIM_SystemComponent is a specialization of the CIM_Component "
    "association that establishes 'part of' relationships between "
    "a System and any ManagedSystemElements of which it is "
    "composed. \n"
    "The use of this association is cautioned - versus the use "
    "of a subclass such as SystemDevice, or a peer association such "
    "as HostedService. This class is very broadly defined which can "
    "lead to erroneous use. For example, Access Points that are "
    "dependent on (and hosted on) a System are NOT Components of the "
    "System. The System is not made up of any AccessPoint 'parts', "
    "which is why a Dependency association, HostedAccessPoint, was "
    "defined. Similarly, a PhysicalPackage is not a 'part' of a "
    "System, since the physical element exists independently of any "
    "internal components, software, etc. In fact, again, a Dependency "
    "relationship is true - where a ComputerSystem is Dependent on "
    "its packaging, as described by the ComputerSystemPackage "
    "association.") ]
class CIM_SystemComponent : CIM_Component {

    [Override ("GroupComponent"), Aggregate,
        Description ("The parent System in the Association.") ]
    CIM_System REF GroupComponent;

    [Override ("PartComponent"), Description (
        "The child element that is a component of a System.") ]
    CIM_ManagedSystemElement REF PartComponent;
};

// ==================================================================
// AdminDomain
// ==================================================================
[Version ("2.6.0"), Description (
    "This is a special grouping of ManagedSystemElements. The "
    "grouping is viewed as a single entity, reflecting that all "
    "of its components are administered similarly - either by "
    "the same user, group of users, or policy. It serves as "
    "an aggregation point to associate one or more of the "
    "following elements: network devices, such as routers and "
    "switches, servers, and other resources that can be "
    "accessed by end systems. This grouping of devices "
    "plays an essential role in ensuring that the same "
    "administrative policy and actions are applied to all "
    "of the devices in the grouping. The specific behavior "
    "and/or semantics of the AdminDomain can be identified "
    "through its aggregated and associated entities. \n") ]
class CIM_AdminDomain : CIM_System {

    [Override ("NameFormat"), Description (
        "The NameFormat property identifies how the Name of the "
        "AdminDomain is generated, using the heuristic specified "
        "in the CIM Version 2 System Model spec. It assumes that the "
        "documented rules are traversed in order, to determine and "
        "assign a Name. The NameFormat Values list defines the "
        "precedence order for assigning the Name of the "
        "AdminDomain."),
        ValueMap {"Other", "AS", "NAP", "NOC", "POP", "RNP", "IP",
            "IPX", "SNA", "Dial", "WAN", "LAN", "ISDN", "Frame Relay",
            "ATM", "E.164", "IB", "FC", "Policy Repository"},
        Values {"Other", "Autonomous System",
            "Network Access Provider", "Network Operations Center",
            "Point of Presence", "Regional Network Provider",
            "IP", "IPX", "SNA", "Dial", "WAN", "LAN", "ISDN",
            "Frame Relay", "ATM", "E.164", "Infiniband",
            "Fibre Channel", "Policy Repository"} ]
    string NameFormat;
};

// ==================================================================
// LogicalDevice
// ==================================================================
[Abstract, Version ("2.7.0"), Description (
    "An abstraction or emulation of a hardware entity, that may "
    "or may not be realized in physical hardware. Any "
    "characteristics of a LogicalDevice that are used to manage "
    "its operation or configuration are contained in, or "
    "associated with, the LogicalDevice object. Examples of the "
    "operational properties of a Printer would be paper sizes "
    "supported, or detected errors. Examples of the configuration "
    "properties of a Sensor Device would be threshold settings. "
    "Various configurations could exist for a LogicalDevice. "
    "These configurations could be contained in Setting objects "
    "and associated with the LogicalDevice.") ]
class CIM_LogicalDevice : CIM_EnabledLogicalElement {

    [Propagated("CIM_System.CreationClassName"),
        Key, MaxLen (256), Description (
            "The scoping System's CreationClassName.") ]
    string SystemCreationClassName;

    [Propagated("CIM_System.Name"),
        Key, MaxLen (256), Description (
            "The scoping System's Name.") ]
    string SystemName;

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or "
        "the subclass used in the creation of an instance. When "
        "used with the other key properties of this class, this "
        "property allows all instances of this class and its "
        "subclasses to be uniquely identified.") ]
    string CreationClassName;

    [Key, MaxLen (64), Description (
        "An address or other identifying information to uniquely "
        "name the LogicalDevice.") ]
    string DeviceID;

    [Description (
        "The primary availability and status of the Device. "
        "(Additional status information can be specified using the "
        "AdditionalAvailability array property.) For example, the "
        "Availability property indicates that the Device is running "
        "and has full power (value=3), or is in a warning (4), test "
        "(5), degraded (10), or power save state (values 13-15 and "
        "17). Regarding the Power Save states, these are defined as "
        "follows: Value 13 (\"Power Save - Unknown\") indicates "
        "that the Device is known to be in a power save mode, but its "
        "exact status in this mode is unknown; 14 (\"Power Save - Low "
        "Power Mode\") indicates that the Device is in a power save "
        "state but still functioning, and may exhibit degraded "
        "performance; 15 (\"Power Save - Standby\") describes that "
        "the Device is not functioning but could be brought to full "
        "power 'quickly'; and value 17 (\"Power Save - Warning\") "
        "indicates that the Device is in a warning state, though also "
        "in a power save mode."),
        ValueMap {"1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11", "12", "13", "14", "15", "16", "17", "18",
            "19", "20", "21"},
        Values {"Other", "Unknown", "Running/Full Power", "Warning",
            "In Test", "Not Applicable", "Power Off", "Off Line",
            "Off Duty", "Degraded", "Not Installed", "Install Error",
            "Power Save - Unknown", "Power Save - Low Power Mode",
            "Power Save - Standby", "Power Cycle",
            "Power Save - Warning", "Paused", "Not Ready",
            "Not Configured", "Quiesced"},
        MappingStrings {"MIF.DMTF|Operational State|006.5",
            "MIB.IETF|HOST-RESOURCES-MIB.hrDeviceStatus",
            "MIF.DMTF|Host Device|001.5"},
        ModelCorrespondence {
            "CIM_LogicalDevice.AdditionalAvailability"} ]
    uint16 Availability;

    [Description (
        "Additional availability and status of the Device beyond that "
        "specified in the Availability property. The Availability "
        "property denotes the primary status and availability of the "
        "Device. In some cases, this will not be sufficient to denote "
        "the complete status of the Device. In those cases, the "
        "AdditionalAvailability property can be used to provide "
        "further information. For example, a Device's primary "
        "Availability may be \"Off line\" (value=8), but it may also "
        "be in a low power state (AdditonalAvailability value=14), or "
        "the Device could be running Diagnostics "
        "(AdditionalAvailability value=5, \"In Test\")."),
        ValueMap {"1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11", "12", "13", "14", "15", "16", "17", "18",
            "19", "20", "21"},
        Values {"Other", "Unknown", "Running/Full Power", "Warning",
            "In Test", "Not Applicable", "Power Off", "Off Line",
            "Off Duty", "Degraded", "Not Installed", "Install Error",
            "Power Save - Unknown", "Power Save - Low Power Mode",
            "Power Save - Standby", "Power Cycle",
            "Power Save - Warning", "Paused", "Not Ready",
            "Not Configured", "Quiesced"},
        ModelCorrespondence {"CIM_LogicalDevice.Availability"} ]
    uint16 AdditionalAvailability[];
```

```
    [ArrayType ("Indexed"), Description (
        "An array of free-form strings providing explanations "
        "and details behind the entries in the OtherIdentifyingInfo "
        "array. Note, each entry of this array is related to the "
        "entry in OtherIdentifyingInfo that is located at the same "
        "index."),
        ModelCorrespondence {"CIM_LogicalDevice.OtherIdentifyingInfo"}]
    string IdentifyingDescriptions[];

    [MaxLen (256), ArrayType ("Indexed"), Description (
        "OtherIdentifyingInfo captures additional data, beyond "
        "DeviceID information, that could be used to identify a "
        "LogicalDevice. One example would be to hold the Operating"
        "System's user friendly name for the Device in this "
        "property."),
        ModelCorrespondence {
            "CIM_LogicalDevice.IdentifyingDescriptions"} ]
    string OtherIdentifyingInfo[];

    [Description (
        "Requests a reset of the LogicalDevice. The return value "
        "should be 0 if the request was successfully executed, "
        "1 if the request is not supported, and some other value "
        "if an error occurred. In a subclass, the set of possible "
        "return codes could be specified, using a ValueMap qualifier "
        "on the method. The strings to which the ValueMap contents "
        "are 'translated' may also be specified in the subclass as a "
        "Values array qualifier.") ]
    uint32 Reset();
};

// ==================================================================
// SystemDevice
// ==================================================================
[Association, Aggregation, Composition, Version ("2.7.0"),
    Description (
        "LogicalDevices may be aggregated by a System. This "
        "relationship is made explicit by the SystemDevice "
        "association.") ]
class CIM_SystemDevice : CIM_SystemComponent {

    [Override ("GroupComponent"), Aggregate, Max (1), Min (1),
        Description ("The parent system in the Association.") ]
    CIM_System REF GroupComponent;

    [Override ("PartComponent"), Weak, Description (
        "The LogicalDevice that is a component of a System.") ]
    CIM_LogicalDevice REF PartComponent;
};

// ==================================================================
// Service
// ==================================================================
[Abstract, Version ("2.7.0"), Description (
    "A CIM_Service is a Logical Element that contains the "
    "information necessary to represent and manage the "
    "functionality provided by a Device and/or SoftwareFeature. "
    "A Service is a general-purpose object to configure and "
    "manage the implementation of functionality. It is not the "
    "functionality itself.") ]
class CIM_Service : CIM_EnabledLogicalElement {

    [Propagated ("CIM_System.CreationClassName"), Key, MaxLen (256),
        Description (
            "The scoping System's CreationClassName.") ]
    string SystemCreationClassName;

    [Propagated ("CIM_System.Name"), Key, MaxLen (256),
        Description (
            "The scoping System's Name.") ]
    string SystemName;

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Override ("Name"), Key, MaxLen (256), Description (
        "The Name property uniquely identifies the Service and "
        "provides an indication of the functionality that is "
        "managed. This functionality is described in more detail in "
        "the object's Description property.") ]
    string Name;

    [MaxLen (64), Write, Description (
        "The name of the primary owner for the service, if "
        "one is defined. The primary owner is the initial "
        "support contact for the Service."),
        MappingStrings {"MIF.DMTF|General Information|001.3"} ]
    string PrimaryOwnerName;

    [MaxLen (256), Write, Description (
        "A string that provides information on how the primary "
        "owner of the Service can be reached (e.g., phone "
        "number, email address, etc.)."),
        MappingStrings {"MIF.DMTF|General Information|001.4"} ]
    string PrimaryOwnerContact;

    [Description (
        "Started is a boolean indicating whether the Service "
        "has been started (TRUE), or stopped (FALSE).") ]
    boolean Started;

    [Description (
        "The StartService method places the Service in the started "
        "state. Note that this method's function overlaps with the "
        "RequestedState property. RequestedState was added to "
        "the model to maintain a record (i.e., a persisted value) "
        "of the last state request. Invoking the StartService "
        "method should set the RequestedState property appropriately."
        "The method returns an integer value of 0 if the Service was "
        "successfully started, 1 if the request is not supported, and "
        "any other number to indicate an error. In a subclass, the "
        "set of possible return codes could be specified using a "
        "ValueMap qualifier on the method. The strings to which the "
        "ValueMap contents are 'translated' may also be specified in "
        "the subclass as a Values array qualifier.") ]
    uint32 StartService();

    [Description (
        "The StopService method places the Service in the stopped "
        "state. Note that this method's function overlaps with the "
        "RequestedState property. RequestedState was added to "
        "the model to maintain a record (i.e., a persisted value) "
        "of the last state request. Invoking the StopService "
        "method should set the RequestedState property appropriately."
        "The method returns an integer value of 0 if the Service was "
        "successfully stopped, 1 if the request is not supported, and "
        "any other number to indicate an error. In a subclass, the "
        "set of possible return codes could be specified, using a "
        "ValueMap qualifier on the method. The strings to which the "
        "ValueMap contents are 'translated' may also be specified in "
        "the subclass as a Values array qualifier.") ]
    uint32 StopService();
};

// ===================================================================
// HostedService
// ===================================================================
[Association, Version ("2.6.0"), Description (
    "CIM_HostedService is an association between a Service and "
    "the System on which the functionality resides. The "
    "cardinality of this association is one-to-many. A System may "
    "host many Services. Services are weak with respect to their "
    "hosting System. Heuristic: A Service is hosted on the "
    "System where the LogicalDevices or SoftwareFeatures that "
    "implement the Service are located. The model does not "
    "represent Services hosted across multiple systems. This is "
    "modeled as an ApplicationSystem that acts as an aggregation "
    "point for Services, that are each located on a single "
    "host.") ]
class CIM_HostedService : CIM_Dependency {

    [Override ("Antecedent"), Max (1), Min (1),
        Description ("The hosting System.") ]
    CIM_System REF Antecedent;

    [Override ("Dependent"), Weak,
        Description ("The Service hosted on the System.") ]
    CIM_Service REF Dependent;
};

// ==================================================================
// ServiceAvailableToElement
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "CIM_ServiceAvailableToElement conveys the semantics "
    "of a Service being available for the 'use' of a ManagedElement. "
    "An example is that a Processor and an enclosure "
    "(a PhysicalElement) may use AlertOnLAN Services to signal an "
    "incomplete or erroneous boot. In reality, AlertOnLAN is simply a "
    "HostedService on a computer system that is generally 'available "
    "for use', and not a dependency of the processor or enclosure. To "
    "describe that use of this service is restricted or has limited "
    "availability/applicability, the CIM_ServiceAvailableToElement "
    "association would be instantiated between the Service and "
    "specific CIM_LogicalDevices and CIM_Chassis.") ]
class CIM_ServiceAvailableToElement {

    [Key, Description (
        "The Service that is available.") ]
    CIM_Service REF ServiceProvided;

    [Key, Description (
        "The ManagedElement that may use the Service.") ]
    CIM_ManagedElement REF UserOfService;
};

// ==================================================================
// ServiceComponent
// ==================================================================
[Association, Aggregation, Version ("2.6.0"), Description (
    "The ServiceComponent aggregation models a set of "
    "subordinate Services that are aggregated together to form "
    "a higher-level service.") ]
class CIM_ServiceComponent : CIM_Component {
    [Override ("GroupComponent"), Aggregate, Description (
        "The parent Service.") ]
    CIM_Service REF GroupComponent;

    [Override ("PartComponent"), Description (
        "The component Service.") ]
   CIM_Service REF PartComponent;
};

// ==================================================================
// ServiceAccessPoint
// ==================================================================
[Abstract, Version ("2.6.0"), Description (
    "CIM_ServiceAccessPoint represents the ability to utilize or "
    "invoke a Service. Access points represent that a Service is "
    "made available to other entities for use.") ]
class CIM_ServiceAccessPoint : CIM_EnabledLogicalElement {
    [Propagated ("CIM_System.CreationClassName"), Key, MaxLen (256),
        Description (
            "The scoping System's CreationClassName.") ]
    string SystemCreationClassName;

    [Propagated ("CIM_System.Name"), Key, MaxLen (256),
        Description (
            "The scoping System's Name.") ]
    string SystemName;

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Override ("Name"), Key, MaxLen (256), Description (
        "The Name property uniquely identifies the ServiceAccessPoint "
        "and provides an indication of the functionality that is "
        "managed. This functionality is described in more detail in "
        "the object's Description property.") ]
    string Name;
};

// ==================================================================
// HostedAccessPoint
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "CIM_HostedAccessPoint is an association between a Service"
    "AccessPoint and the System on which it is provided. The "
    "cardinality of this association is one-to-many and is weak "
    "with respect to the System. Each System may host many "
    "ServiceAccessPoints. Heuristic: If the implementation of "
    "the ServiceAccessPoint is modeled, it must be "
    "implemented by a Device or SoftwareFeature that is part of "
    "the System hosting the ServiceAccessPoint.") ]
class CIM_HostedAccessPoint : CIM_Dependency {

    [Override ("Antecedent"), Max (1), Min (1),
        Description ("The hosting System.") ]
    CIM_System REF Antecedent;

    [Override ("Dependent"), Weak,
        Description ("The SAP(s) that are hosted on this System.") ]
    CIM_ServiceAccessPoint REF Dependent;
};

// ==================================================================
// ServiceAccessBySAP
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "CIM_ServiceAccessBySAP is an association that identifies the "
    "access points for a Service. For example, a printer may be "
    "accessed by Netware, MacIntosh, or Windows ServiceAccess"
    "Points, potentially hosted on different Systems.") ]
class CIM_ServiceAccessBySAP : CIM_Dependency {

    [Override ("Antecedent"),
        Description ("The Service.") ]
    CIM_Service REF Antecedent;

    [Override ("Dependent"), Description (
        "An Access Point for a Service. Access points are dependent "
        "in this relationship since they have no function without a "
        "corresponding Service.") ]
    CIM_ServiceAccessPoint REF Dependent;
};

// ==================================================================
// SAPSAPDependency
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "CIM_SAPSAPDependency is an association between a Service "
    "AccessPoint and another ServiceAccessPoint indicating that "
    "the latter is required in order for the former ServiceAccess"
    "Point to utilize or connect with its Service. For example, "
    "to print at a network printer, local Print Access Points "
    "must utilize underlying network-related SAPs, or "
    "ProtocolEndpoints, in order to send the print request.") ]
class CIM_SAPSAPDependency : CIM_Dependency {

    [Override ("Antecedent"), Description (
        "The required ServiceAccessPoint.") ]
    CIM_ServiceAccessPoint REF Antecedent;

    [Override ("Dependent"), Description (
        "The ServiceAccessPoint that is dependent on an underlying "
        "SAP.") ]
    CIM_ServiceAccessPoint REF Dependent;
};

// ==================================================================
// ActiveConnection
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "This association defines a connection that is currently "
    "communicating, or is configured to communicate, "
    "between two ServiceAccessPoints. It is used when the "
    "connection is not treated as a ManagedElement itself "
    "(i.e., with Status, statistics, etc.), but its "
    "existence is known. The SAPs that are connected are "
    "typically at the same networking or application layer.") ]
class CIM_ActiveConnection : CIM_SAPSAPDependency {

    [Override ("Antecedent"), Description (
        "A ServiceAccessPoint that is configured to communicate "
        "and/or is actively communicating with the Dependent "
        "SAP. In a unidirectional connection, this is the "
        "SAP which is transmitting.") ]
    CIM_ServiceAccessPoint REF Antecedent;

    [Override ("Dependent"), Description (
        "A second SeriveAccessPoint that is/can communicate "
        "with the Antecedent SAP. In a unidirectional "
        "connection, this is the SAP that is receiving the "
        "communication.") ]
    CIM_ServiceAccessPoint REF Dependent;

    [Description (
        "TRUE means that this connection is unidirectional; FALSE "
        "means that this connection is bidirectional. When the "
        "connection is unidirectional, the 'speaker' should be "
        "defined as the Antecedent reference. In a bidirectional "
        "connection, the selection of which AccessPoint is the "
        "Antecedent or Dependent is immaterial.") ]
    boolean IsUnidirectional;
};

// ==================================================================
// RemoteServiceAccessPoint
// ==================================================================
[Version ("2.7.0"), Description (
    "RemoteServiceAccessPoint describes access and/or "
    "addressing information for a remote connection, that is "
    "known to a 'local' network element. This information is "
    "scoped/contained by the 'local' network element, since "
    "this is the context in which it is 'remote'. \n\n"
    "Why the remote access point is relevant and information "
    "on its use are described by subclassing RemoteService"
    "AccessPoint, or by associating to it.") ]
class CIM_RemoteServiceAccessPoint : CIM_ServiceAccessPoint {

    [Description (
        "Access and/or addressing information for a remote "
        "connection. This can be a host name, network "
        "address, or similar information."),
        ModelCorrespondence {
            "CIM_RemoteServiceAccessPoint.InfoFormat"} ]
    string AccessInfo;

    [Description (
        "An enumerated integer describing the format and "
        "interpretation of the AccessInfo property."),
        ValueMap {"1", "2", "3", "4", "5", "6", "7", "8",
            "9", "10..99", "100", "101", "102", "103", "104",
            "105..199", "200", "32768..65535"},
        Values {"Other", "Host Name", "Ipv4 Address",
            "Ipv6 Address", "IPX Address", "DECnet Address",
            "SNA Address", "Autonomous System Number",
            "MPLS Label", "DMTF Reserved", "Dial String",
            "Ethernet Address", "Token Ring Address",
            "ATM Address", "Frame Relay Address",
            "DMTF Reserved", "URL", "Vendor Specific"},
        ModelCorrespondence {
            "CIM_RemoteServiceAccessPoint.OtherInfoFormatDescription"}]
    uint16 InfoFormat;

    [Description (
        "Describes the format when the property InfoFormat "
        "is set to 1 (\"Other\")."),
        ModelCorrespondence {
            "CIM_RemoteServiceAccessPoint.InfoFormat"} ]
    string OtherInfoFormatDescription;
};

// ==================================================================
// RemotePort
// ==================================================================
[Version ("2.7.0"), Description (
    "RemotePort adds port information to the access data "
    "(such as IP Address), specified in and inherited from "
    "RemoteServiceAccessPoint. For example, using the "
    "additional properties in this class, one could indicate that "
    "UDP Port 123 is accessed at IP Address xyz. The IP Address "
    "would be specified in the inherited AccessInfo property, "
    "while the UDP Port number would be specified in this "
    "class' property, PortInfo.") ]
class CIM_RemotePort : CIM_RemoteServiceAccessPoint {

    [Description (
        "Port information that may further specify the "
        "remote access information."),
        ModelCorrespondence {
            "CIM_RemotePort.PortProtocol"} ]
    string PortInfo;

    [Description (
        "An enumerated integer describing the protocol "
        "of the port addressed by PortInformation."),
        ValueMap {"1", "2", "3", "32768..65535"},
        Values {"Other", "TCP", "UDP", "Vendor Specific"},
        ModelCorrespondence {
            "CIM_RemotePort.OtherProtocolDescription"} ]
    uint16 PortProtocol;

    [Description (
        "Describes the protocol when the property PortProtocol "
        "is set to 1 (\"Other\")."),
        ModelCorrespondence {
            "CIM_RemotePort.PortProtocol"} ]
    string OtherProtocolDescription;
};
```

```
// ==================================================================
// ServiceAccessURI
// ==================================================================
[Version ("2.7.0"), Description (
    "CIM_ServiceAccessURI represents the ability to utilize or "
    "invoke a Service via a uniform resource identifier. "
    "IETF RFC 1738 describes the Uniform Resource Locator which is a "
    "more specific form of a URI, and IETF RFC 2396 describes the "
    "broader, inclusive URI. Quoting from the RFC: 'A Uniform "
    "Resource Identifier (URI) is a compact string of characters for "
    "identifying an abstract or physical resource.' In the context "
    "of a ServiceAccessPoint, a URI contains the information "
    "necessary to access an associated service (associated via "
    "ServiceAccessBySAP), using a syntax specific to that service.") ]
class CIM_ServiceAccessURI : CIM_ServiceAccessPoint {

    [Required, MaxLen (256), Description (
        "The LabeledURI property is based on IETF RFC 2079 and its "
        "value and syntax is service specific.") ]
    string LabeledURI;
};

// ==================================================================
// ProtocolEndpoint
// ==================================================================
[Version ("2.7.0"), Description (
    "A communication point from which data may be sent or "
    "received. ProtocolEndpoints link system/computer interfaces "
    "to LogicalNetworks.") ]
class CIM_ProtocolEndpoint : CIM_ServiceAccessPoint {

    [Override ("Name"), MaxLen(256), Description (
        "A string which identifies this ProtocolEndpoint with either "
        "a port or an interface on a device. To ensure uniqueness, "
        "the Name property should be prepended or appended with "
        "information from the Type or OtherTypeDescription "
        "properties. The method chosen is described in the "
        "NameFormat property of this class.") ]
    string Name;

    [MaxLen (256), Description (
        "NameFormat contains the naming heuristic that is chosen to "
        "ensure that the value of the Name property is unique. For "
        "example, one might choose to prepend the name of the port "
        "or interface with the Type of ProtocolEndpoint that this "
        "instance is (e.g., IPv4) followed by an underscore.") ]
    string NameFormat;

    [Description (
        "ProtocolType is an enumeration providing information to "
        "categorize and classify different instances of this class. "
        "For most instances, information in this enumeration and the "
        "subclass' definition overlap. However, there are several "
        "cases where a specific subclass of ProtocolEndpoint is not "
        "required (for example, there is no Fibre Channel Endpoint). "
        "Hence, this property is needed to define the type of "
        "Endpoint. Note that in a future major release of the CIM "
        "Schema, it is likely that ProtocolEndpoint will be ABSTRACT, "
        "forcing subclassing and removing the need for this "
        "property."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11", "12", "13", "14", "15", "16", "17", "18",
            "19", "20", "21", "22", "23", "24", "25", "26", "27"},
        Values {"Unknown", "Other", "IPv4", "IPv6", "IPX",
            "AppleTalk", "DECnet", "SNA", "CONP", "CLNP",
            "VINES", "XNS", "ATM", "Frame Relay",
            "Ethernet", "TokenRing", "FDDI", "Infiniband",
            "Fibre Channel", "ISDN BRI Endpoint",
            "ISDN B Channel Endpoint", "ISDN D Channel Endpoint",
            // 22
            "IPv4/v6", "BGP", "OSPF", "MPLS", "UDP", "TCP"},
        ModelCorrespondence {
            "CIM_ProtocolEndpoint.OtherTypeDescription"} ]
    uint16 ProtocolType;

    [MaxLen(64), Description (
        "A string describing the type of ProtocolEndpoint when "
        "the Type property of this class (or any of its "
        "subclasses) is set to 1 (i.e., \"Other\"). This "
        "property should be set to NULL when the Type property "
        "is any value other than 1."),
        ModelCorrespondence {"CIM_ProtocolEndpoint.ProtocolType"} ]
    string OtherTypeDescription;
};

// ==================================================================
// BindsTo
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "This association establishes a ServiceAccessPoint "
    "as a requestor of protocol services from a "
    "ProtocolEndpoint. Typically, this association runs "
    "between SAPs and endpoints on a single system. Since a "
    "ProtocolEndpoint is a kind of ServiceAccessPoint, this "
    "binding can be used to establish a layering of two "
    "protocols, the upper layer represented by the Dependent "
    "and the lower layer represented by the Antecedent.") ]
class CIM_BindsTo : CIM_SAPSAPDependency {

    [Override ("Antecedent"), Description (
        "The lower-level endpoint which is accessed by the SAP.") ]
    CIM_ProtocolEndpoint REF Antecedent;

    [Override ("Dependent"), Description (
        "The AccessPoint or ProtocolEndpoint dependent on the "
        "lower level endpoint.") ]
    CIM_ServiceAccessPoint REF Dependent;
};
```

```
// ===================================================================
// SettingData
// ===================================================================
[Abstract, Version ("2.7.0"), Description (
    "The SettingData class represents configuration-related and "
    "operational parameters for one or more ManagedElement(s). A "
    "ManagedElement may have multiple SettingData objects associated "
    "with it. The current operational values for an Element's "
    "parameters are reflected by properties in the Element itself or "
    "by properties in its associations. These properties do not have "
    "to be the same values present in the SettingData object. For "
    "example, a modem may have a SettingData baud rate of 56Kb/sec "
    "but be operating at 19.2Kb/sec.") ]
class CIM_SettingData : CIM_ManagedElement {

    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this class. "
        "In order to ensure uniqueness within the NameSpace, the "
        "value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "    <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked, "
        "or otherwise unique name that is owned by the business "
        "entity creating/defining the InstanceID, or is a registered "
        "ID that is assigned to the business entity by a recognized "
        "global authority (This is similar to the "
        "<Schema Name>_<Class Name> structure of Schema class names.) "
        "In addition, to ensure uniqueness <OrgID> MUST NOT contain a "
        "colon (':'). When using this algorithm, the first colon to "
        "appear in InstanceID MUST appear between <OrgID> and "
        "<LocalID>. \n"
        "<LocalID> is chosen by the business entity and SHOULD not be "
        "reused to identify different underlying (real-world) "
        "elements. If the above 'preferred' algorithm is not used, "
        "the defining entity MUST assure that the resultant "
        "InstanceID is not reused across any InstanceIDs produced by "
        "this or other providers for this instance's NameSpace. \n"
        "For DMTF-defined instances, the 'preferred' algorithm MUST "
        "be used with the <OrgID> set to 'CIM'.") ]
    string InstanceID;

    [Override ("ElementName"), Required, Description (
        "The user friendly name for this instance of SettingData. "
        "In addition, the user friendly name can be used as an "
        "index property for a search of query. (Note that Name "
        "does not have to be unique within a namespace.)") ]
    string ElementName;
};
```

```
// ===================================================================
// ElementSettingData
// ===================================================================
[Association, Version ("2.7.0"), Description (
    "ElementSettingData represents the association between "
    "ManagedElements and applicable setting data. This "
    "association also describes whether this is a default "
    "or current setting.") ]
class CIM_ElementSettingData {

    [Key, Description (
        "The managed element.") ]
    CIM_ManagedElement REF ManagedElement;

    [Key, Description (
        "The SettingData object associated with the element.") ]
    CIM_SettingData REF SettingData;

    [Description (
        "An enumerated integer indicating that the referenced "
        "setting is a default setting for the element, or that "
        "this information is unknown."),
        ValueMap {"0", "1", "2"},
        Values {"Unknown", "Is Default", "Is Not Default"} ]
    uint16 IsDefault;

   [Description (
        "An enumerated integer indicating that the referenced "
        "setting is currently being used in the operation "
        "of the element, or that this information is unknown."),
    ValueMap {"0", "1", "2"},
    Values {"Unknown", "Is Current", "Is Not Current"} ]
    uint16 IsCurrent;
};

// ==================================================================
// Profile
// ==================================================================
[Version ("2.7.0"), Description (
    "Profile specifies a general collection of Settings/SettingData "
    "that are logically grouped together. The settings are  "
    "'collected' into the Profile using the MemberOfCollection "
    "association. Note that a Profile obtains its setting data and "
    "other necessary information from collected Setting/SettingData "
    "instances, and by the definition of additional properties in "
    "subclasses. One example of a Profile is to define the collection "
    "of settings for a particular person. Another is to define the "
    "collection of settings appropriate when initializing a system. \n"
    "\n"
    "Profile is tied to a ManagedElement using the ElementProfile "
    "association. The ManagedElement provides context for the "
    "Profile, and the settings collected by Profile SHOULD be "
    "settings of the associated ManagedElement. Note that a "
    "ManagedElement MAY NOT exist when the Profile is "
    "instantiated. This is true when the Profile is used in the "
    "creation of the real-world entity represented by (and then "
    "instantiated in) the ManagedElement.") ]
class CIM_Profile : CIM_Collection {

    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this class. "
        "In order to ensure uniqueness within the NameSpace, the "
        "value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "    <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked, "
        "or otherwise unique name that is owned by the business "
        "entity creating/defining the InstanceID, or is a registered "
        "ID that is assigned to the business entity by a recognized "
        "global authority (This is similar to the "
        "<Schema Name>_<Class Name> structure of Schema class names.) "
        "In addition, to ensure uniqueness <OrgID> MUST NOT contain a "
        "colon (':'). When using this algorithm, the first colon to "
        "appear in InstanceID MUST appear between <OrgID> and "
        "<LocalID>. \n"
        "<LocalID> is chosen by the business entity and SHOULD not be "
        "reused to identify different underlying (real-world) "
        "elements. If the above 'preferred' algorithm is not used, "
        "the defining entity MUST assure that the resultant "
        "instanceID is not reused across any InstanceIDs produced by "
        "this or other providers for this instance's NameSpace. \n"
        "For DMTF-defined instances, the 'preferred' algorithm MUST "
        "be used with the <OrgID> set to 'CIM'.") ]
    string InstanceID;
};

// ==================================================================
// ElementProfile
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "ElementProfile represents the association between "
    "ManagedElements and the Profiles (collection of "
    "settings) defined for them. This association describes that "
    "the settings collected by the Profile apply to the "
    "associated ManagedElement.") ]
class CIM_ElementProfile {

    [Key, Description (
        "The managed element.") ]
    CIM_ManagedElement REF ManagedElement;

    [Key, Description (
        "The Profile object associated with the element.") ]
    CIM_Profile REF Profile;
};
```

```
// ===================================================================
// end of file
// ===================================================================
```

A.4 Management SAP

```
// ====================================================================
// Title:       Mobile and Directory Schema - ManagementSAP
//              association (and superclass) from CIM Version 2.9
// Filename:    MaD_ManagementSAP11.mof
// Version:     1.1.0
// Date:        11/10/03 (Initial release)
// ====================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ====================================================================

#pragma Locale ("en_US")

// ===================================================================
// SAPAvailableForElement
// ===================================================================
[Association, Experimental, Version ("2.7.1000"), Description (
    "CIM_SAPAvailableForElement conveys the semantics of a Service "
    "Access Point being available for a ManagedElement. When "
    "CIM_SAPAvailableForElement is not instantiated, then the SAP is "
    "assumed to be generally available. If instantiated, the SAP "
    "is only available for the associated ManagedElements. For "
    "example, a device might provide management access through a "
    "URL. This association allows the URL to be advertised for the "
    "device.") ]
class CIM_SAPAvailableForElement {

    [Key, Description (
        "The Service Access Point that is available.") ]
    CIM_ServiceAccessPoint REF AvailableSAP;

    [Key, Description (
        "The ManagedElement for which the SAP is available.") ]
    CIM_ManagedElement REF ManagedElement;
};

// ===================================================================
// ManagementSAP
// ===================================================================
[Association, Experimental, Version ("2.8.1000"), Description (
    "Indicates that the associated ServiceAccessPoint provides "
    "management access for the referenced element. For example, "
    "it might be a UDP interface for SNMP access, or a URL.") ]
class CIM_ManagementSAP : CIM_SAPAvailableForElement {

    [Override ("AvailableSAP"), Description (
        "The access point through which the element can be "
        "managed.") ]
    CIM_ServiceAccessPoint REF AvailableSAP;
    [Override ("ManagedElement"),
    Description ("The element being managed.") ]
    CIM_ManagedElement REF ManagedElement;
};

// ==================================================================
// end of file
// ==================================================================
```

A.5 Physical Information

```
// ===================================================================
// Title:       Mobile and Directory Schema - Subset of the CIM
//              Physical Model Version 2.7 + a few properties in
//              Physical Element mapped in CIM Version 2.8 from the
                Entity MIB
// Filename:    MaD_Physical11.mof
// Version:     1.1.0
// Release:     Preliminary
// Date:        11/10/03
// Change History:
//   Feb 2003 - Original release
//   Nov 2003 - Move to CIM Version 2.7 Final Release classes
// ===================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ===================================================================

#pragma locale ("en_US")

// =================================================================
// PhysicalElement
// ==================================================================
[Abstract, Version ("2.7.1000"), Description (
    "Subclasses of CIM_PhysicalElement define any component of a "
    "System that has a distinct physical identity. Instances of "
    "this class can be defined in terms of labels that can be "
    "physically attached to the object. All Processes, Files, "
    "and LogicalDevices are considered not to be Physical "
    "Elements. For example, it is not possible to attach a "
    "label to a modem. It is only possible to attach a label to "
    "the card that implements the modem. The same card could "
    "also implement a LAN adapter. These are tangible Managed "
    "SystemElements (usually actual hardware items) that have a "
    "physical manifestation of some sort. A ManagedSystem "
    "Element is not necessarily a discrete component. For "
    "example, it is possible for a single Card (which is a type "
    "of PhysicalElement) to host more than one LogicalDevice. "
    "The card would be represented by a single PhysicalElement "
    "associated with multiple Devices.") ]
class CIM_PhysicalElement : CIM_ManagedSystemElement {

    [Key, MaxLen (256), Description (
        "An arbitrary string that uniquely identifies the Physical "
        "Element and serves as the Element's key. The Tag property "
        "can contain information such as asset tag or serial number "
        "data. The key for PhysicalElement is placed very high in the "
        "object hierarchy in order to independently identify the "
        "hardware/entity, regardless of physical placement in or on "
        "Cabinets, Adapters, etc. For example, a hotswappable or "
        "removeable component may be taken from its containing "
        "(scoping) Package and be temporarily unused. The object "
        "still continues to exist - and may even be inserted into a "
        "different scoping container. Therefore, the key for Physical "
        "Element is an arbitrary string and is defined independently "
        "of any placement or location-oriented hierarchy.") ]
    string Tag;

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Override ("ElementName"),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalName"} ]
    string ElementName;

    [MaxLen (256), Description (
        "The name of the organization responsible for producing the "
        "PhysicalElement. This may be the entity from whom the "
        "Element is purchased, but this is not necessarily true."),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalMfgName"} ]
    string Manufacturer;

    [MaxLen (256), Description (
        "The name by which the PhysicalElement is generally known."),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalDescr"} ]
    string Model;

    [MaxLen (64), Description (
        "The stock keeping unit number for this PhysicalElement.") ]
    string SKU;

    [MaxLen (256), Description (
        "A manufacturer-allocated number used to identify the "
        "Physical Element."),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalSerialNum"} ]
    string SerialNumber;

    [MaxLen (64), Description (
        "A string indicating the version of the PhysicalElement."),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalHardwareRev"} ]
    string Version;

    [MaxLen (256), Description (
        "The part number assigned by the organization responsible for "
        "producing or manufacturing the PhysicalElement."),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalModelName"} ]
    string PartNumber;

    [Write, Description (
        "OtherIdentifyingInfo captures additional data beyond that of "
```

```
        "Tag information that could be used to identify a Physical "
        "Element. One example is barcode data associated with an "
        "Element that also has an asset tag. Note that if only bar "
        "code data is available and is unique/able to be used as an "
        "Element key, this property would be NULL and the barcode "
        "data used as the class key, in the Tag property."),
        MappingStrings {"MIB.IETF|Entity-MIB.entPhysicalAlias"} ]
    string OtherIdentifyingInfo;

    [Description (
        "Boolean indicating that the PhysicalElement is powered on "
        "(TRUE), or is currently off (FALSE).") ]
    boolean PoweredOn;

    [Description (
        "Date that this PhysicalElement was manufactured.") ]
    datetime ManufactureDate;

    [Experimental, Description (
        "A vendor-specific hardware type for the PhysicalElement. "
        "It describes the specific equipment type for the element, "
        "as defined by the vendor/manufacturer."),
    MappingStrings { "MIB.IETF|Entity-MIB.entPhysicalVendorType"} ]
    string VendorEquipmentType;

    [Experimental, Write, Description (
        "User-assigned and writeable asset tracking identifier for "
        "the PhysicalElement."),
    MappingStrings { "MIB.IETF|Entity-MIB.entPhysicalAssetID"} ]
    string UserTracking;

    [Experimental, Description (
        "Boolean indicating whether this PhysicalElement can be "
        "FRUed (TRUE) or not (FALSE)."),
    MappingStrings { "MIB.IETF|Entity-MIB.entPhysicalIsFRU"} ]
    boolean CanBeFRUed;
};

// ==================================================================
// Realizes
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "CIM_Realizes is the association that defines the mapping "
    "between LogicalDevices and the PhysicalElements that "
    "implement them.") ]
class CIM_Realizes : CIM_Dependency {

    [Override ("Antecedent"), Description (
        "The physical component that implements the Device.") ]
    CIM_PhysicalElement REF Antecedent;

    [Override ("Dependent"), Description (
        "The LogicalDevice.") ]
    CIM_LogicalDevice REF Dependent;
```

```
};

// ===================================================================
// Location
// ===================================================================
[Version ("2.6.0"), Description (
    "The Location class specifies the position and address of "
    "a PhysicalElement.") ]
class CIM_Location : CIM_ManagedElement {

    [Key, MaxLen (256),
        Description (
            "Name is a free-form string defining a label for the "
            "Location. It is a part of the key for the object.") ]
    string Name;

    [Key, MaxLen (256), Description (
        "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.") ]
    string PhysicalPosition;

    [MaxLen (1024), Description (
        "Address is a free-form string indicating a street, building, "
        "or other type of address for the PhysicalElement's "
        "Location.") ]
    string Address;
};

// ==================================================================
// ElementLocation
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "ElementLocation associates a ManagedElement with "
    "a Location for positioning, inventory, maintenance, and similar "
    "purposes. PhysicalElements can certainly have Locations - "
    "this is explicitly defined in a subclass, PhysicalElement"
    "Location. However, other ManagedElements may also be associated "
    "with Locations. For example, Organizations may be 'in' one or "
    "more Locations, or Services may be restricted to a Location.") ]
class CIM_ElementLocation {

    [Key, Description (
        "The ManagedElement whose Location is specified.") ]
    CIM_ManagedElement REF Element;

    [Key, Description ("The element's Location.") ]
    CIM_Location REF PhysicalLocation;
};
```

```
// ===================================================================
// ContainedLocation
// ===================================================================
[Association, Aggregation, Composition,
    Version ("2.7.0"), Description (
        "ContainedLocation defines one location in the context "
        "of another. For example, a campus may 'contain' a building, "
        "which in turn 'contains' a floor.") ]
class CIM_ContainedLocation : CIM_Component {

    [Aggregate, Override ("GroupComponent"), Max (1),
        Description ("The containing location.") ]
    CIM_Location REF GroupComponent;

    [Override ("PartComponent"),
        Description ("The contained location.") ]
    CIM_Location REF PartComponent;
};

// ===================================================================
// SystemPackaging
// ===================================================================
[Association, Version("2.7.0"), Description (
    "Similar to the way that LogicalDevices are 'Realized' by "
    "PhysicalElements, Systems may be associated with specific "
    "packaging/PhysicalElements. This association explicitly "
    "defines the relationship between a System and its packaging.") ]
class CIM_SystemPackaging : CIM_Dependency {

    [Override ("Antecedent"), Description (
        "The PhysicalElement(s) that provide the packaging of a "
        "System.") ]
    CIM_PhysicalElement REF Antecedent;

    [Override ("Dependent"),
        Description ("The System whose packaging is described.") ]
    CIM_System REF Dependent;
};

// ===================================================================
// PhysicalPackage
// ===================================================================
[Version ("2.6.0"), Description (
    "The PhysicalPackage class represents PhysicalElements that "
    "contain or host other components. Examples are a Rack "
    "enclosure or an adapter Card.") ]
class CIM_PhysicalPackage : CIM_PhysicalElement {
    [Description (
        "A PhysicalPackage is Removable if it is designed 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.") ]
    boolean Removable;

    [Description (
        "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.") ]
    boolean Replaceable;

    [Description (
        "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 (i.e., 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.") ]
    boolean HotSwappable;

    [Description (
        "The height of the PhysicalPackage in inches."),
        Units ("Inches") ]
    real32 Height;

    [Description (
        "The depth of the PhysicalPackage in inches."),
        Units ("Inches") ]
    real32 Depth;

    [Description (
        "The width of the PhysicalPackage in inches."),
        Units ("Inches") ]
    real32 Width;

    [Description (
        "The weight of the PhysicalPackage in pounds."),
        Units ("Pounds") ]
    real32 Weight;
};
```

```
// ==================================================================
// Container
// ==================================================================
[Association, Aggregation, Version ("2.6.0"), Description (
    "The Container association represents the relationship "
    "between a contained and a containing PhysicalElement. "
    "A containing object must be a PhysicalPackage.") ]
class CIM_Container : CIM_Component {

    [Override ("GroupComponent"), Aggregate, Max (1), Description (
        "The PhysicalPackage that contains other PhysicalElements, "
        "including other Packages."),
        MappingStrings {
            "MIB.IETF|Entity-MIB.entPhysicalContainedIn"} ]
    CIM_PhysicalPackage REF GroupComponent;

    [Override ("PartComponent"), Description (
        "The PhysicalElement which is contained in the Package.") ]
    CIM_PhysicalElement REF PartComponent;

    [Description (
        "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 CIM_Location object."),
        MappingStrings {
            "MIB.IETF|Entity-MIB.entPhysicalParentRelPos"} ]
    string LocationWithinContainer;
};

// ==================================================================
// PhysicalFrame
// ==================================================================
[Version ("2.7.0"), Description (
    "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." ) ]
class CIM_PhysicalFrame : CIM_PhysicalPackage {

    [Description ("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.") ]
    string CableManagementStrategy;

    [Description (
        "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."),
        ArrayType ("Indexed"),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8"},
        Values {"Unknown", "Other", "Service From Top",
            "Service From Front", "Service From Back",
            "Service From Side", "Sliding Trays",
            "Removable Sides", "Moveable"},
        ModelCorrespondence {"CIM_PhysicalFrame.ServiceDescriptions"} ]
    uint16 ServicePhilosophy[];

    [Description ("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."),
        ArrayType ("Indexed"),
        ModelCorrespondence {"CIM_PhysicalFrame.ServicePhilosophy"} ]
    string ServiceDescriptions[];

    [Description ("Boolean indicating whether the Frame is protected "
        "with a lock.") ]
    boolean LockPresent;

    [Description ("Boolean indicating whether the Frame is equipped "
        "with an audible alarm.") ]
    boolean AudibleAlarm;

    [Description ("Boolean indicating that the equipment includes "
        "a visible alarm.") ]
    boolean VisibleAlarm;

    [Description ("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."),
        ValueMap {"1", "2", "3", "4", "5"},
        Values {"Other", "Unknown", "No Breach", "Breach Attempted",
            "Breach Successful"},
        MappingStrings {
            "MIF.DMTF|Physical Container Global Table|004.12"},
        ModelCorrespondence {"CIM_PhysicalFrame.BreachDescription"} ]
    uint16 SecurityBreach;

    [Description ("BreachDescription is a free-form string providing "
        "more information if the SecurityBreach property indicates "
        "that a breach or some other security-related event "
        "occurred."),
        ModelCorrespondence {"CIM_PhysicalFrame.SecurityBreach"} ]
    string BreachDescription;
```

```
    [Description (
        "Boolean indicating that the Frame is currently locked.") ]
    boolean IsLocked;
};

// ==================================================================
// Rack
// ==================================================================
[Version ("2.6.0"), Description (
    "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.") ]
class CIM_Rack : CIM_PhysicalFrame {

    [Override ("Height"), Description (
        "The height of the PhysicalPackage in 'U's. 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."),
        Units ("Us") ]
    real32 Height;

    [Description ("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 Country "
        "Designation property."),
        ValueMap {"0", "1", "2", "3", "4"},
        Values {"Unknown", "Standard 19 Inch", "Telco",
            "Equipment Shelf", "Non-Standard"},
        ModelCorrespondence {"CIM_Rack.CountryDesignation"} ]
    uint16 TypeOfRack;

    [Description ("Designation of the country for which the Rack "
        "is designed. Country code strings are as defined by "
        "ISO/IEC 3166. The rack type is specified in the TypeOf "
        "Rack property."),
        ModelCorrespondence {"CIM_Rack.TypeOfRack"} ]
    string CountryDesignation;
};

// ==================================================================
// Chassis
// ==================================================================
[Version ("2.7.0"), Description (
    "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.") ]
class CIM_Chassis : CIM_PhysicalFrame {

    [Description (
        "An enumerated, integer-valued array indicating the type of "
        "Chassis."),
        ArrayType ("Indexed"),
        ValueMap {"1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
            "20", "21", "22", "23", "24", "25"},
        Values {"Other", "Unknown", "Desktop", "Low Profile Desktop",
               "Pizza Box", "Mini Tower", "Tower", "Portable",
               "LapTop",
               // 10
               "Notebook", "Hand Held", "Docking Station",
               "All in One", "Sub Notebook", "Space-Saving",
               "Lunch Box", "Main System Chassis",
               "Expansion Chassis", "SubChassis",
               // 20
               "Bus Expansion Chassis", "Peripheral Chassis",
               "Storage Chassis", "Rack Mount Chassis",
               "Sealed-Case PC", "Multi-system Chassis"},
        MappingStrings {
            "MIF.DMTF|Physical Container Global Table|004.1"},
        ModelCorrespondence {"CIM_Chassis.TypeDescriptions"} ]
    uint16 ChassisTypes[];

    [Description (
        "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."),
        ArrayType ("Indexed"),
        ModelCorrespondence {"CIM_Chassis.ChassisTypes"} ]
    string TypeDescriptions[];
};

// ==================================================================
// Card
// ==================================================================
[Version ("2.6.0"), Description (
    "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 CIM_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.") ]
class CIM_Card : CIM_PhysicalPackage {

    [Description (
        "Boolean indicating that this Card is a Motherboard or, "
        "more generically, a baseboard in a Chassis.") ]
    boolean HostingBoard;

    [Description (
        "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.") ]
    string SlotLayout;

    [Description (
        "Boolean indicating that at least one daughterboard or "
        "auxiliary Card is required in order to function properly.") ]
    boolean RequiresDaughterBoard;

    [Description ("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."),
        ModelCorrespondence{"CIM_Card.RequirementsDescription"} ]
    boolean SpecialRequirements;

    [Description ("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."),
        ModelCorrespondence{"CIM_Card.SpecialRequirements"} ]
    string RequirementsDescription;
};

// ==================================================================
// end of file
// ==================================================================
```

A.6 Logical Information

```
// ==================================================================
// Title:       Mobile and Directory Schema - Subset of the CIM Core,
//              System, Device, and Networks Models Version 2.7, and
//              new classes for user preference settings,
//              DHCP management, and wireless information
// Filename:    MaD_Logical11.mof
// Version:     1.1.0
// Release:     Preliminary
// Date:        11/10/03
// Change History:
//   Feb 2003 - Original release
//   Nov 2003 - Move to CIM Version 2.7 Final Release classes
// ==================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ==================================================================

#pragma locale ("en_US")

// ==================================================================
// ComputerSystem
// ==================================================================
[Version ("2.7.0"), Description (
    "A class derived from System that is a special collection of "
    "ManagedSystemElements. This collection provides "
    "compute capabilities and serves as aggregation point to "
    "associate one or more of the following elements: FileSystem, "
    "OperatingSystem, Processor, and Memory (Volatile and/or "
    "NonVolatile Storage).") ]
class CIM_ComputerSystem : CIM_System {

    [Override ("NameFormat"), Description (
        "The ComputerSystem object and its derivatives are Top "
        "Level Objects of CIM. They provide the scope for numerous "
        "components. Having unique System keys is required. "
        "The NameFormat property identifies how the ComputerSystem "
        "Name is generated. The NameFormat ValueMap qualifier "
        "defines the various mechanisms for assigning the name. "
        "Note that another name can be assigned and "
        "used for the ComputerSystem that better suit "
        "a business, using the inherited ElementName property."),
        ValueMap {"Other", "IP", "Dial", "HID", "NWA", "HWA", "X25",
            "ISDN", "IPX", "DCC", "ICD", "E.164", "SNA", "OID/OSI",
            "WWN"} ]
    string NameFormat;

    [MaxLen (256), ArrayType ("Indexed"), Description (
        "OtherIdentifyingInfo captures additional data, beyond "
        "System Name information, that could be used to identify "
        "a ComputerSystem. One example would be to hold the "
        "Fibre Channel World-Wide Name (WWN) of a node. Note that "
        "if only the Fibre Channel name is available and is "
        "unique (able to be used as the System key), then this "
        "property would be NULL and the WWN would become the "
        "System key, its data placed in the Name property."),
        ModelCorrespondence {
            "CIM_ComputerSystem.IdentifyingDescriptions"} ]
    string OtherIdentifyingInfo[];

    [ArrayType ("Indexed"), Description (
        "An array of free-form strings providing explanations "
        "and details behind the entries in the OtherIdentifying "
        "Info array. Note, each entry of this array is related "
        "to the entry in OtherIdentifyingInfo that is located at "
        "the same index."),
        ModelCorrespondence {
            "CIM_ComputerSystem.OtherIdentifyingInfo"} ]
    string IdentifyingDescriptions[];

    [Description (
        "Enumeration indicating whether the ComputerSystem is "
        "a special-purpose System (i.e., dedicated to a particular "
        "use), versus being 'general purpose'. For example, one "
        "could specify that the System is dedicated to \"Print\" "
        "(value=11) or acts as a \"Hub\" (value=8). \n"
        "A clarification is needed with respect to the value "
        "17 (\"Mobile User Device\"). An example of a dedicated "
        "user device is a mobile phone or a barcode scanner in a "
        "store that communicates via radio frequency. These systems "
        "are quite limited in functionality and programmability, "
        "and are not considered 'general purpose' computing "
        "platforms. "
        "Alternately, an example of a mobile system that is "
        "'general purpose' (i.e., is NOT dedicated) is a hand-held "
        "computer. Although limited in its programmability, new "
        "software can be downloaded and its functionality expanded "
        "by the user."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11", "12", "13", "14", "15", "16", "17"},
        Values {"Not Dedicated", "Unknown", "Other", "Storage",
            "Router", "Switch", "Layer 3 Switch",
            "Central Office Switch", "Hub", "Access Server",
            "Firewall", "Print", "I/O", "Web Caching", "Management",
            "Block Server", "File Server", "Mobile User Device"},
        ModelCorrespondence {
            "CIM_ComputerSystem.OtherDedicatedDescriptions"} ]
    uint16 Dedicated[];

    [Description (
        "A string describing how or why the system is dedicated "
        "when the Dedicated array includes the value 2, "
        "\"Other\"."),
        ModelCorrespondence {"CIM_ComputerSystem.Dedicated"} ]
    string OtherDedicatedDescriptions[];
```

```
};

// ==================================================================
// SoftwareIdentity
// ==================================================================
[Version ("2.7.0"), Description (
    "SoftwareIdentity represents software, viewed as an asset "
    "and/or individually identifiable entity (similar to Physical"
    "Element). It does NOT indicate whether the software is "
    "installed, executing, etc. (The latter is the role of the "
    "SoftwareFeature/SoftwareElement classes and the Application "
    "Model.) Since software may be acquired, SoftwareIdentity can be "
    "associated with a Product using the ProductSoftwareComponent "
    "relationship. \n"
    "Note that the Application Model manages the deployment and "
    "installation of software via the classes, SoftwareFeatures and "
    "SoftwareElements. The deployment/installation concepts are "
    "related to the asset/identity one. In fact, a SoftwareIdentity "
    "may correspond to a Product, or to one or more SoftwareFeatures "
    "or SoftwareElements - depending on the granularity of these "
    "classes and the deployment model. The correspondence of Software"
    "Identity to Product, SoftwareFeature, or SoftwareElement is "
    "indicated using the ConcreteIdentity association. Note that "
    "there may not be sufficient detail or instrumentation to "
    "instantiate ConcreteIdentity. And, if the association is "
    "instantiated, some duplication of information may result. For "
    "example, the Vendor described in the instances of Product and "
    "SoftwareIdentity MAY be the same. However, this is not "
    "necessarily true, and it is why vendor and similar information "
    "are duplicated in this class. \n"
    "Note that ConcreteIdentity can also be used to describe "
    "the relationship of the software to any LogicalFiles "
    "that result from installing it. As above, there may not be "
    "sufficient detail or instrumentation to instantiate this "
    "association.") ]
class CIM_SoftwareIdentity : CIM_LogicalElement {

    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this class. "
        "In order to ensure uniqueness within the NameSpace, the "
        "value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "    <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked, "
        "or otherwise unique name that is owned by the business "
        "entity creating/defining the InstanceID, or is a registered "
        "ID that is assigned to the business entity by a recognized "
        "global authority (This is similar to the "
        "<Schema Name>_<Class Name> structure of Schema class names.) "
        "In addition, to ensure uniqueness <OrgID> MUST NOT contain a "
        "colon (':'). When using this algorithm, the first colon to "
        "appear in InstanceID MUST appear between <OrgID> and "
```

```
            "<LocalID>. \n"
            "<LocalID> is chosen by the business entity and SHOULD not be "
            "reused to identify different underlying (real-world) "
            "elements. If the above 'preferred' algorithm is not used, "
            "the defining entity MUST assure that the resultant "
            "InstanceID is not reused across any InstanceIDs produced by "
            "this or other providers for this instance's NameSpace. \n"
            "For DMTF-defined instances, the 'preferred' algorithm MUST "
            "be used with the <OrgID> set to 'CIM'.") ]
        string InstanceID;

        [Description (
            "The major number component of the software's version "
            "information - for example, '12' from version "
            "12.1(3)T. This property is defined as a numeric value "
            "to allow the determination of 'newer' versus 'older' "
            "releases. A 'newer' major release is indicated by a larger "
            "numeric value.") ]
        uint16 MajorVersion;

        [Description (
            "The minor number component of the software's version "
            "information - for example, '1' from version "
            "12.1(3)T. This property is defined as a numeric value "
            "to allow the determination of 'newer' versus 'older' "
            "releases. A 'newer' minor release is indicated by a larger "
            "numeric value.") ]
        uint16 MinorVersion;

        [Description (
            "The revision or maintenance release component of the "
            "software's version information - for example, '3' from "
            "version 12.1(3)T. This property is defined as a numeric "
            "value to allow the determination of 'newer' versus 'older' "
            "releases. A 'newer' revision is indicated by a larger "
            "numeric value.") ]
        uint16 RevisionNumber;

        [Description (
            "The build number of the software.") ]
        uint16 BuildNumber;

        [Description (
            "A string representing the complete software version "
            "information - for example, '12.1(3)T'. This string and "
            "the numeric major/minor/revision/build properties are "
            "complementary. Since vastly different representations and "
            "semantics exist for versions, it is not assumed that "
            "one representation is sufficient to permit a client to "
            "perform computations (i.e., the values are numeric) and "
            "a user to recognize the software's version (i.e., the "
            "values are understandable and readable). Hence, both "
            "numeric and string representations of version are provided.")
    ]
        string VersionString;

        [Description (
            "Specifies the target operating systems of the software. "
            "Due to the extreme variability in operating systems, "
            "this property is defined as a string array.") ]
        string TargetOperatingSystems[];

        [Description (
            "Manufacturer of this software.") ]
        string Manufacturer;

        [Description (
            "The language editions supported by the software. The "
            "language codes defined in ISO 639 should be used. ") ]
        string Languages[];

        [Description (
            "An array of enumerated integers that classify this software. "
            "For example, the software could be instrumentation (value=5) "
            "or firmware and diagnostic software (6 and 7)."),
            ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"},
            Values {"Unknown", "Other", "Driver",
                "Configuration Software", "Application Software",
                "Instrumentation", "Firmware/BIOS",
                "Diagnostic Software", "Operating System", "Middleware"},
            ArrayType ("Indexed"),
            ModelCorrespondence {
                "CIM_SoftwareIdentity.ClassificationDescriptions"} ]
        uint16 Classifications[];

        [Description (
            "An array of free-form strings providing more detailed "
            "explanations for any of the entries in the Classifications "
            "array. Note that each entry is related to one in the "
            "Classifications array located at the same index."),
            ArrayType ("Indexed"),
            ModelCorrespondence {"CIM_SoftwareIdentity.Classifications"} ]
        string ClassificationDescriptions[];
    };

    // ==================================================================
    // LANEndpoint
    // ==================================================================
        [Version ("2.7.0"), Description (
            "A communication endpoint which, when its associated "
            "interface device is connected to a LAN, may send and "
            "receive data frames. LANEndpoints include Ethernet, "
            "Token Ring, and FDDI interfaces.") ]
    class CIM_LANEndpoint : CIM_ProtocolEndpoint {

        [Description (
            "A label or identifier for the LAN Segment to which the "
            "Endpoint is connected. If the Endpoint is not currently "
            "active/connected or this information is not known, then "
            "LANID is NULL.") ]
        string LANID;

        [MaxLen(12), Description (
            "The principal unicast address used in communication "
            "with the LANEndpoint. The MAC address is formatted as "
            "twelve hexadecimal digits (e.g., \"010203040506\"), "
            "with each pair representing one of the six octets "
            "of the MAC address in \"canonical\" bit order "
            "according to RFC 2469.") ]
        string MACAddress;

        [Description (
            "Other unicast addresses that may be used to communicate "
            "with the LANEndpoint.") ]
        string AliasAddresses[];

        [Description (
            "Multicast addresses to which the LANEndpoint listens.") ]
        string GroupAddresses[];

        [Description (
            "The largest information field that may be sent or received "
            "by the LANEndpoint."), Units ("Bits") ]
        uint32 MaxDataSize;
    };

    // ==================================================================
    // IPProtocolEndpoint
    // ==================================================================
        [Version ("2.7.0"), Description (
            "A ProtocolEndpoint that is dedicated to running IP.") ]
    class CIM_IPProtocolEndpoint : CIM_ProtocolEndpoint {

        [Description (
            "The IPv4 address that this ProtocolEndpoint represents.") ]
        string IPv4Address;

        [Description (
            "The IPv6 address that this ProtocolEndpoint represents.") ]
        string IPv6Address;

        [Description (
            "The mask for the IPv4 address of this ProtocolEndpoint, "
            "if one is defined.") ]
        string SubnetMask;

        [Description (
            "The prefix length for the IPv6 address of this Protocol"
            "Endpoint, if one is defined.") ]
        uint8 PrefixLength;
        [Description (
            "This property explicitly defines support for different "
            "versions of the IP protocol, for this Endpoint."),
            ValueMap {"0", "1", "2", "3"},
            Values {"Unknown", "IPv4 Only", "IPv6 Only",
            "Both IPv4 and IPv6"} ]
        uint16 IPVersionSupport;
    };

    // ======================================================================
    // ChallengeUserPreferences
    // ======================================================================
    [Experimental, Version ("2.8.0"), Description (
        "A subclass of SettingData representing a user's preference "
        "regarding publishing their location information. This "
        "information is tied to a UserEntity using the "
        "ElementSettingData association.") ]
    class MaD_ChallengeUserPreferences : CIM_SettingData {

        [Description (
            "A boolean indicating the user's choice whether their "
            "physical location may be advertised (TRUE) or kept private "
            "(FALSE). By default, this property is TRUE.") ]
        boolean AdvertiseLocation = TRUE;
    };

    // ======================================================================
    // LogicalPort
    // ======================================================================
    [Version ("2.7.0"), Description (
        "The abstraction of a port or connection point of a Device. "
        "This object should be instantiated when the Port has "
        "independent management characteristics from the Device that "
        "includes it. Examples are a Fibre Channel Port and a USB Port.") ]
    class CIM_LogicalPort : CIM_LogicalDevice {

        [Description ("The bandwidth of the Port in Bits per Second."),
            Units ("Bits per Second") ]
        uint64 Speed;

        [Description (
            "The maximum bandwidth of the Port in Bits per Second."),
            Units ("Bits per Second") ]
        uint64 MaxSpeed;
    };

    // ======================================================================
    // PortOnDevice
    // ======================================================================
    [Association, Version ("2.6.0"), Description (
        "PortOnDevice associates a Port or connection point with its "
        "Device.") ]
    class CIM_PortOnDevice : CIM_Dependency {
        [Override ("Antecedent"),
            Description ("The Device that includes the Port.") ]
        CIM_LogicalDevice REF Antecedent;

        [Override ("Dependent"),
            Description ("The Port on the Device.") ]
        CIM_LogicalPort REF Dependent;
    };

    // ===================================================================
    // NetworkPort
    // ===================================================================
    [Version ("2.7.0"), Description (
        "NetworkPort is the logical representation of network "
        "communications hardware - a physical connector and the "
        "setup/operation of the network chips, at the lowest layers "
        "of a network stack.") ]
    class CIM_NetworkPort : CIM_LogicalPort {

        [Override ("Speed"), Description (
            "The current bandwidth of the Port in Bits per Second. "
            "For ports that vary in bandwidth or for those where "
            "no accurate estimation can be made, this property should "
            "contain the nominal bandwidth."),
            Units ("Bits per Second"),
            MappingStrings {"MIB.IETF|MIB-II.ifSpeed",
                "MIF.DMTF|Network Adapter 802 Port|001.5"} ]
        uint64 Speed;

        [Description (
            "PortType is defined to force consistent naming of the 'type' "
            "property in subclasses and to guarantee unique enum values "
            "for all instances of NetworkPort. When set to 1 (\"Other\"), "
            "related property OtherPortType contains a string description "
            "of the port's type. A range of values, DMTF_Reserved, "
            "has been defined that allows subclasses to override and "
            "define their specific port types."),
            ValueMap {"0", "1", "2", "3..15999", "16000..65535"},
            Values {"Unknown", "Other", "Not Applicable", "DMTF Reserved",
                "Vendor Reserved"},
            ModelCorrespondence {"CIM_NetworkPort.OtherNetworkPortType"} ]
        uint16 PortType;

        [Description (
            "Describes the type of module, when PortType is "
            "set to 1 (\"Other\")."),
            ModelCorrespondence {"CIM_NetworkPort.PortType"} ]
        string OtherNetworkPortType;

        [Description (
            "NetworkPorts are often numbered relative to either a logical "
            "modules or a network element.") ]
        uint16 PortNumber;
        [Description (
            "An enumeration of the types of links. When set to 1 "
            "(\"Other\"), the related property OtherLinkTechnology "
            "contains a string description of the link's type."),
            ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8"},
            Values {"Unknown", "Other", "Ethernet", "IB", "FC",
                "FDDI", "ATM", "Token Ring", "Frame Relay"},
            ModelCorrespondence {"CIM_NetworkPort.OtherLinkTechnology"} ]
        uint16 LinkTechnology;

        [Description (
            "A string value describing LinkTechnology when it "
            "is set to 1, \"Other\"."),
            ModelCorrespondence {"CIM_NetworkPort.LinkTechnology"} ]
        string OtherLinkTechnology;

        [MaxLen (64), Description (
            "PermanentAddress defines the network address hardcoded into "
            "a port. This 'hardcoded' address may be changed via "
            "firmware upgrade or software configuration. If so, this "
            "field should be updated when the change is made. "
            "PermanentAddress should be left blank if no 'hardcoded' "
            "address exists for the NetworkAdapter."),
            MappingStrings {"MIF.DMTF|Network Adapter 802 Port|001.2"} ]
        string PermanentAddress;

        [MaxLen (64), Description (
            "An array of strings indicating the network addresses for "
            "the port."),
            MappingStrings {"MIF.DMTF|Network Adapter 802 Port|001.3"} ]
        string NetworkAddresses[];

        [Description (
            "Boolean indicating that the port is operating in "
            "full duplex mode.") ]
        boolean FullDuplex;

        [Description (
            "A boolean indicating whether the NetworkPort is capable "
            "of automatically determining the speed or other "
            "communications characteristics of the attached network "
            "media.") ]
        boolean AutoSense;

        [Description (
            "The maximum transmission unit (MTU) that can be supported."),
            Units ("Bytes") ]
        uint64 SupportedMaximumTransmissionUnit;

        [Description (
            "The active or negotiated maximum transmission unit (MTU) "
            "that can be supported."),
            Units ("Bytes") ]
        uint64 ActiveMaximumTransmissionUnit;
```

```
};

// ==================================================================
// WirelessPort
// ==================================================================
[Experimental, Version ("2.8.0"), Description (
    "Capabilities and management of a wireless port.") ]
class CIM_WirelessPort : CIM_NetworkPort {

    [Override("PortType"), Description (
        "The type of port." ),
        ValueMap {"0", "1", "75", "76", "77", "78", "79", "80",
            "16000.." },
        Values {"Unknown", "Other", "Client Station", "Repeater",
            "Access Point", "Bridge Host", "Bridge", "Bridge Root",
            "Vendor Reserved"} ]
    uint16 PortType;

    [Description (
        "The current signal strength represented as a percentage. "
        "For example, if the current strength is 3 bars out of 5, "
        "then this property would be set to 60."),
        Units ("Percentage") ]
    uint8 SignalStrength;
};

// ==================================================================
// WirelessLANEndpoint
// ==================================================================
[Experimental, Version ("2.8.0"), Description (
    "A ProtocolEndpoint supporting 802.11 wireless networking.") ]
class CIM_WirelessLANEndpoint : CIM_ProtocolEndpoint {

    [Override ("Name"), Description (
        "The MAC Address of the wireless endpoint.") ]
    string Name;

    [MaxLen (32), Description (
        "A 32-character identifier (the Service Set Identifier) "
        "that is inserted into the header of packets transmitted "
        "from the endpoint. This identifier helps to distinguish "
        "one Wireless LAN from another.") ]
    string SSID;

    [Description (
        "Boolean indicating whether the Wired Equivalent Privacy "
        "algorithm is enabled for communications on this endpoint. By "
        "default, this property is set to FALSE.") ]
    boolean WEPEnabled = FALSE;

    [Description (
        "Boolean indicating whether WEP key mixing is used. "
        "By default, this property is set to FALSE.") ]
    boolean WEPKeyMixEnabled = FALSE;
```

```
};

// ===================================================================
// DHCPService
// ===================================================================
[Experimental, Version ("2.8.0"), Description (
    "This class represents the functionality provided by a DHCP "
    "Server, to assign IP addresses to clients, according to "
    "administrator preferences and definitions. This class is "
    "associated via ElementSettingData to an administrator's address "
    "and option assignment settings, or uses policy to assign "
    "addresses.") ]
class CIM_DHCPService : CIM_Service {
};

// ===================================================================
// DHCPAddress
// ===================================================================
[Experimental, Version ("2.8.0"), Description (
    "DHCPAddress describes the status of individual IP addresses "
    "managed by the DHCPService. Instances can be associated via "
    "ElementSettingData to an administrator's processing settings "
    "for the address (such as whether a domain name is sent to "
    "the client, whether the DNS server should be updated upon "
    "address assignment, etc.). For ease of use, the AddressLease "
    "association is defined - relating the assigned address(es) "
    "to a client system. If more specific information is desired, "
    "a subclass of LogicalIdentity can be used to associate a "
    "specific DHCPAddress (on a DHCP server), with an "
    "IProtocolEndpoint (on a client system).") ]
class CIM_DHCPAddress : CIM_ProtocolEndpoint {

    [Override ("Name"), Description (
        "The IP Address of the endpoint.") ]
    string Name;

    [Description (
        "The state of the DHCP address within the context of the DHCP "
        "server."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11", "12"},
        Values {"Unknown", "Other", "Reserved", "Free", "Active",
            "Assigned", "Unassigned", "Not Assignable", "Expired",
            "Released", "Reset", "Abandoned", "Backup"} ]
    uint16 AddressState[];

    [Description (
        "If the address is assigned or in use, the expiration time "
        "of the lease.") ]
    datetime ExpirationTime;
};
```

```
// ===================================================================
// DHCPServiceManagesAddress
// ===================================================================
[Association, Version ("2.8.0"), Description (
    "This relationship defines the DHCPAddresses managed by a "
    "particular DHCP server.") ]
class CIM_DHCPServiceManagesAddress : CIM_Dependency {

    [Override ("Antecedent"), Min (1), Max (1),
        Description ("The DHCP server.") ]
    CIM_DHCPService REF Antecedent;

    [Override ("Dependent"),
        Description (
        "The addresses managed/assigned by the DHCP server.") ]
    CIM_DHCPAddress REF Dependent;
};

// ===================================================================
// AddressLease
// ===================================================================
[Association, Version ("2.8.0"), Description (
    "This relationship defines the DHCPAddresses assigned to a "
    "client system.") ]
class CIM_AddressLease : CIM_Dependency {

    [Override ("Antecedent"),
        Description ("The IP address that is assigned to the client.")]
    CIM_DHCPAddress REF Antecedent;

    [Override ("Dependent"), Max (1),
        Description (
        "The client system dependent on the address assigned.") ]
    CIM_ComputerSystem REF Dependent;
};

// ===================================================================
// end of file
// ===================================================================
```

A.7 Interoperability Information

```
// ===================================================================
// Title:       Mobile and Directory Schema - Subset of CIM Interop
//              Version 2.7 and addition of Directory-related classes
// Filename:    MaD_Interop11.mof
// Version:     1.1.0
// Release:     Preliminary
// Date:        11/10/03
// Change History:
//   Feb 2003 - Original release
//   Nov 2003 - Move to CIM Version 2.7 Final Release classes
// ===================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ===================================================================

#pragma Locale ("en_US")

// ==================================================================
// WBEMService
// ==================================================================
[Abstract, Version ("2.6.0"), Description (
    "An abstract class for WBEM services such as the "
    "ObjectManager (CIM Server), Providers, protocol adapters, "
    "or other pluggable components. A type of CIM_Service "
    "that provides associated capabilities and details about the "
    "communication capabilities, namespaces, etc. for a specific "
    "WBEMService.") ]
class CIM_WBEMService : CIM_Service {
};

// ==================================================================
// ObjectManager
// ==================================================================
[Version ("2.7.0"), Description (
    "A type of CIM_Service that defines the capabilities of the "
    "CIM Server in which this ObjectManager class resides. Details "
    "related to communicating with the ObjectManager, and the "
    "Manager's basic capabilities, are stored in instances of the "
    "associated CommunicationMechanism class available through the "
    "CommMechanismForManager association. It is assumed that "
    "Basic Read operations must be supported by all "
    "ObjectManager's in order to retrieve any additional "
    "detail.") ]
class CIM_ObjectManager : CIM_WBEMService {
};

// ==================================================================
// ObjectManagerCommunicationMechanism
// ==================================================================
[Version ("2.7.0"), Description (
    "The class, ObjectManagerCommunicationMechanism, describes "
    "access to an ObjectManager. It describes a protocol and "
    "data encoding that can be used for communication. When all "
    "instances of this class are enumerated for an ObjectManager "
    "(using the CommMechanismForManager association), all possible "
    "protocol and encoding schemes will be known. Also, specific "
    "capabilities (for example, basic read or query) that are "
    "supported in the protocol/encoding are described - using the "
    "ProfilesSupported property.") ]
class CIM_ObjectManagerCommunicationMechanism : CIM_ServiceAccessPoint{

    [Required, Description (
        "CommunicationMechanism describes an encoding and protocol "
        "which can be used to communicate with the ObjectManager. "
        "At this time, only one encoding and protocol are "
        "standardized by the DMTF - 2 \"CIM-XML\". If this is "
        "supported by an ObjectManager, the specified value should be "
        "indicated. In the future, other 'standard' mechanisms may be "
        "defined. In addition, a vendor specific encoding/protocol "
        "value may be specified by using the value 1, \"Other\", and "
        "defining the mechanism in the "
        "OtherComminicationMechanismDescription property."),
        ValueMap {"0", "1", "2"},
        Values {"Unknown", "Other", "CIM-XML"},
        ModelCorrespondence {
            "CIM_ObjectManagerCommunicationMechanism."
            "OtherCommunicationMechanismDescription"} ]
    uint16 CommunicationMechanism;

    [Description (
        "A free-form string providing a description of the "
        "supported encoding and protocol when 1, \"Other\", is "
        "specified in CommunicationMechanism."),
        ModelCorrespondence {
    "CIM_ObjectManagerCommunicationMechanism.CommunicationMechanism"} ]
    string OtherCommunicationMechanismDescription;

    [Required, Description (
        "Enumerated array describing the types of operations "
        "supported by the ObjectManager, using the encoding/protocol "
        "specified in the property, CommunicationMechanism. "
        "The enumeration is based on the Functional Profiles defined "
        "for conformance in the DMTF document, Specification for CIM "
        "Operations over HTTP."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"},
        Values {"Unknown", "Other", "Basic Read", "Basic Write",
           "Schema Manipulation", "Instance Manipulation",
           "Association Traversal", "Query Execution",
           "Qualifier Declaration", "Indications"},
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_ObjectManagerCommunicationMechanism."
            "FunctionalProfileDescriptions"} ]
    uint16 FunctionalProfilesSupported[];
```

```
[Description (
    "Free-form strings providing descriptions of the supported "
    "operations of the object manager. Entries in the array "
    "are correlated with those in the ProfilesSupported array. "
    "An entry in this Descriptions array MUST be provided when "
    "1, \"Other\", is specified in the ProfilesSupported array."),
    ArrayType ("Indexed"),
    ModelCorrespondence {
        "CIM_ObjectManagerCommunicationMechanism."
        "FunctionalProfilesSupported"} ]
string FunctionalProfileDescriptions[];

[Required, Description (
    "Boolean indicating whether the ObjectManager supports "
    "multiple operation requests (TRUE) or only simple requests "
    "(FALSE).") ]
boolean MultipleOperationsSupported;

[Required, Description (
    "Enumerated array describing the types of authentication "
    "supported by the ObjectManager, using the encoding/protocol "
    "specified in the property, CommunicationMEchanism. "
    "The defined values represent the authentication defined in "
    "the DMTF document, Specification for CIM Operations over "
    "HTTP."),
    ValueMap {"0", "1", "2", "3", "4"},
    Values {"Unknown", "Other", "None", "Basic", "Digest"},
    ArrayType ("Indexed"),
    ModelCorrespondence {
        "CIM_ObjectManagerCommunicationMechanism."
        "AuthenticationMechanismDescriptions"} ]
uint16 AuthenticationMechanismsSupported[];

[Description (
    "Free-form strings providing descriptions of the "
    "supported mechanisms. Entries in this array are correlated "
    "with those in the AuthenticationMechanismsSupported array. "
    "An entry in this Descriptions array MUST be provided when "
    "1, \"Other\", is specified in "
    "AuthenticationMechanismsSupported."),
    ArrayType ("Indexed"),
    ModelCorrespondence {
        "CIM_ObjectManagerCommunicationMechanism."
        "AuthenticationMechanismsSupported"} ]
string AuthenticationMechanismDescriptions[];

[Required, Description (
    "Provides the protocol version for this service "
    "access point. Version information MUST be in "
    "the form of M.N, where M is a numeric that describes "
    "the Major version and N is a numeric that describes "
    "the minor version.") ]
string Version;
```

```
};

// ==================================================================
// CIMXMLCommunicationMechanism
// ==================================================================
[Version ("2.6.0"), Description (
    "This class specializes ObjectManagerCommunicationMechanism, "
    "adding properties specific to the CIM-XML protocol (XML "
    "encoding and CIM Operations).") ]
class CIM_CIMXMLCommunicationMechanism :
        CIM_ObjectManagerCommunicationMechanism {

    [Override ("CommunicationMechanism"), Required]
    uint16 CommunicationMechanism = 2;

    [Override ("Version"), Required, Description (
        "Enumeration describing the CIM-XML protocol version "
        "supported by the ObjectManager."),
        ValueMap {"1.0", "1.1", "1.2"} ]
    string Version;

    [Required, Description (
        "Describes whether the CIM Server is strictly validating "
        "(validates the XML document against the DTD) or not "
        "(loosely validating).") ]
    boolean CIMValidated;
};

// ==================================================================
// CommMechanismForManager
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "CommMechanismForManager is an association between an "
    "ObjectManager and an ObjectManagerCommunicationMechanism "
    "class. The latter describes a possible encoding/protocol/"
    "set of operations for accessing the referenced "
    "ObjectManager.") ]
class CIM_CommMechanismForManager : CIM_ServiceAccessBySAP {

    [Override ("Antecedent"), Min (1), Max (1), Description (
        "The specific ObjectManager whose communication mechanism "
        "is described.") ]
    CIM_ObjectManager REF Antecedent;

    [Override ("Dependent"), Min (1), Description (
        "The encoding/protocol/set of operations that may be used "
        "to communicate with the referenced ObjectManager.") ]
    CIM_ObjectManagerCommunicationMechanism REF Dependent;
};

// ==================================================================
// Namespace
// ==================================================================
[Version ("2.6.0"), Description (
    "Namespace provides a domain (in other words, a container), "
    "in which the instances [of a class] are guaranteed to be "
    "unique per the KEY qualifier definitions. It is named "
    "relative to the CIM_ObjectManager implementation that "
    "provides such a domain.") ]
class CIM_Namespace : CIM_ManagedElement {

    [Propagated("CIM_ObjectManager.SystemCreationClassName"), Key,
        MaxLen (256), Description (
            "The scoping System's CreationClassName.") ]
    string SystemCreationClassName;

    [Propagated("CIM_ObjectManager.SystemName"), Key, MaxLen (256),
        Description ("The scoping System's Name.") ]
    string SystemName;

    [Propagated ("CIM_ObjectManager.CreationClassName"), Key,
        MaxLen (256), Description (
            "The scoping ObjectManager's CreationClassName.") ]
    string ObjectManagerCreationClassName;

    [Propagated ("CIM_ObjectManager.Name"), Key, MaxLen (256),
        Description ("The scoping ObjectManager's Name.") ]
    string ObjectManagerName;

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Key, MaxLen (256), Description (
        "A string to uniquely identify the Namespace within "
        "the ObjectManager.") ]
    string Name;

    [Required, Write, Description (
        "Enumeration indicating the organization/schema of the "
        "Namespace's objects. For example, they may be instances "
        "of classes of a specific CIM version."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "200", "201", "202"},
        Values {"Unknown", "Other", "CIM 1.0", "CIM 2.0",
            "CIM 2.1", "CIM 2.2", "CIM 2.3", "CIM 2.4", "CIM 2.5",
            "CIM 2.6", "CIM 2.7", "DMI Recast", "SNMP Recast",
            "CMIP Recast"},
        ModelCorrespondence {"CIM_Namespace.DescriptionOfClassInfo"} ]
    uint16 ClassInfo;

    [Write, Description (
        "A string providing more detail (beyond the general "
        "classification in ClassInfo) for the object hierarchy of "
```

```
            "the Namespace."),
            ModelCorrespondence {"CIM_Namespace.ClassInfo"} ]
        string DescriptionOfClassInfo;
};

// ===================================================================
// SystemIdentification
// ===================================================================
[Version ("2.7.0"), Description (
    "A Namespace may represent data for one or many systems, "
    "that are local, remote (different than the system on which "
    "the ObjectManager is running) or aggregated. The System"
    "Identification class provides enough data to identify the "
    "system(s) represented in the Namespace. It is weak to "
    "the Namespace.") ]
class CIM_SystemIdentification : CIM_ManagedElement {

    [Propagated("CIM_Namespace.SystemCreationClassName"),Key,
        MaxLen (256), Description (
            "The scoping System's CreationClassName.") ]
    string SystemCreationClassName;

    [Propagated("CIM_Namespace.SystemName"), Key, MaxLen (256),
        Description ("The scoping System's Name.") ]
    string SystemName;

    [Propagated ("CIM_Namespace.ObjectManagerCreationClassName"),
        Key, MaxLen (256), Description (
            "The scoping ObjectManager's CreationClassName.") ]
    string ObjectManagerCreationClassName;

    [Propagated ("CIM_Namespace.ObjectManagerName"), Key,
        MaxLen (256), Description (
            "The scoping ObjectManager's Name.") ]
    string ObjectManagerName;

    [Propagated ("CIM_Namespace.CreationClassName"), Key,
        MaxLen (256), Description (
            "The scoping Namespace's CreationClassName.") ]
    string NamespaceCreationClassName;

    [Propagated ("CIM_Namespace.Name"), Key, MaxLen (256),
        Description ("The scoping Namespace's Name.") ]
    string NamespaceName;

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Key, MaxLen (256), Description (
        "A string uniquely identifying the name of the system "
        "represented in the Namespace.") ]
    string Name;

    [Required, Write, Description (
        "Enumeration indicating the format of the system "
        "identification and/or addressing information."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
            "10", "11"},
        Values {"Unknown", "Other", "Short DNS Name",
            "Fully qualified DNS Name",
            "Windows Domain Name", "NetWare Server Name",
            "UNIX Hostname", "Globally Unique System Identifier",
            "IP Address", "Modem Dialup String", "MAC Address",
            "IPMI Address" },
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_SystemIdentification.IdentificationData",
            "CIM_SystemIdentification.FormatDescriptions"} ]
    uint16 IdentificationFormats[];

    [Write, Description (
        "Strings further describing the format of the system "
        "identification information."),
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_SystemIdentification.IdentificationFormats",
            "CIM_SystemIdentification.IdentificationData"} ]
    string FormatDescriptions[];

    [Write, Description (
        "Strings containing the system identification information. "
        "The format is described by the corresponding array item in "
        "IdentificationFormats."),
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_SystemIdentification.IdentificationFormats",
            "CIM_SystemIdentification.FormatDescriptions"} ]
    string IdentificationData[];
};

// ==================================================================
// NamespaceInManager
// ==================================================================
[Association, Version ("2.7.0"), Description (
    "NamespaceInManager is an association describing the "
    "Namespaces hosted by a CIM ObjectManager.") ]
class CIM_NamespaceInManager : CIM_Dependency {

    [Override ("Antecedent"), Min (1), Max (1), Description (
        "The ObjectManager containing a Namespace.") ]
    CIM_ObjectManager REF Antecedent;

    [Override ("Dependent"), Weak, Description (
        "The Namespace in an ObjectManager.") ]
    CIM_Namespace REF Dependent;
};

// ==================================================================
// SystemInNamespace
// ==================================================================
[Association, Version ("2.6.0"), Description (
    "SystemInNamespace is an association that allows enumeration "
    "of the system(s) represented in a Namespace.") ]
class CIM_SystemInNamespace {

    [Key, Min (1), Max (1), Description (
        "The Namespace containing management objects from "
        "one or more systems.") ]
    CIM_Namespace REF ManagedNamespace;

    [Key, Weak, Description (
        "Identification information for systems in the Namespace.") ]
    CIM_SystemIdentification REF Identification;

    [Required, Write, Description (
        "A list of enumerated values providing a high level "
        "description of the data contained and allowed in the "
        "Namespace. Additional clarification is provided in the "
        "DescriptionOfContainedData array."),
        ArrayType ("Indexed"),
        ValueMap {"0", "1", "2", "3", "4"},
        Values {"Unknown", "Other", "Local System Data",
            "Data Aggregated from Multiple Systems",
            "Proxied Data"},
        ModelCorrespondence {
            "CIM_SystemInNamespace.DescriptionOfContainedData"} ]
    uint16 ScopeOfContainedData[];

    [Write, Description (
        "An array of free-form strings providing more detailed "
        "explanations for the data/objects contained in the "
        "Namespace, as described by the ContainedData array. "
        "Note, each entry of this array is related to the entry "
        "in the ContainedData array that is located at the same "
        "index."),
        ArrayType ("Indexed"),
        ModelCorrespondence {
            "CIM_SystemInNamespace.ScopeOfContainedData"} ]
    string DescriptionOfContainedData[];
};

// ==================================================================
// DirectoryService
// ==================================================================
[Experimental, Version ("2.8.0"), Description (
    "A class describing the management of a directory.") ]
class CIM_DirectoryService : CIM_Service {
};

// ==================================================================
// DirectoryCommunicationMechanism
// ==================================================================
[Experimental, Version ("2.8.0"), Description (
    "The class, DirectoryCommunicationMechanism, describes "
    "access to a directory. It describes the protocol and "
    "data encoding that can be used for communication. When all "
    "instances of this class are enumerated for a Directory "
    "(using the CommMechanismForDirectory association), all possible "
    "protocol and encoding schemes will be known.") ]
class CIM_DirectoryCommunicationMechanism :  CIM_ServiceAccessPoint {

    [Required, Description (
        "CommunicationMechanism describes the encoding and protocol "
        "which can be used to communicate with the DirectoryService."),
        ValueMap {"0", "1", "2", "3"},
        Values {"Unknown", "Other", "LDAP", "DSML"},
        ModelCorrespondence {
            "CIM_DirectoryCommunicationMechanism."
            "OtherCommunicationMechanismDescription"} ]
    uint16 CommunicationMechanism;

    [Description (
        "A free-form string providing a description of the "
        "supported protocols when 1, \"Other\", is specified in "
        "the CommunicationMechanism."),
        ModelCorrespondence {
        "CIM_DirectoryCommunicationMechanism.CommunicationMechanism"} ]
    string OtherCommunicationMechanismDescription;
};

// ==================================================================
// CommMechanismForDirectory
// ==================================================================
[Association, Experimental, Version ("2.8.0"), Description (
    "CommMechanismForDirectory is an association between a "
    "Directory and an instance of DirectoryCommunicationMechanism. "
    "The latter describes a possible encoding and protocol "
    "for accessing the referenced Directory.") ]
class CIM_CommMechanismForDirectory : CIM_ServiceAccessBySAP {

    [Override ("Antecedent"), Min (1), Max (1), Description (
        "The specific Directory whose communication mechanism "
        "is described.") ]
    CIM_DirectoryService REF Antecedent;

    [Override ("Dependent"), Min (1), Description (
        "The encoding and protocol that may be used to "
        "communicate with the referenced Directory.") ]
    CIM_DirectoryCommunicationMechanism REF Dependent;
};
```

```
// ===================================================================
// end of file
// ===================================================================
```

A.8 User Information

```
// ===================================================================
// Title:       Mobile and Directory Schema - Subset of the CIM
//              User and Security Model Version 2.7, and new classes
//              from CIM Version 2.8: UserContact, Identity, and
//              AssignedIdentity
// Filename:    MaD_User11.mof
// Version:     1.1.0
// Status:      Preliminary
// Date:        11/10/03
// Change History:
//   Feb 2003 - Original release
//   Nov 2003 - Move to CIM Version 2.7 Final Release classes
// ===================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ===================================================================

#pragma Locale ("en_US")

// ==================================================================
// OrganizationalEntity
// ==================================================================
[Abstract, Version ("2.6.0"), Description (
    "OrganizationalEntity is an abstract class from which classes "
    "that fit into an organizational structure are derived.") ]
class CIM_OrganizationalEntity : CIM_ManagedElement {
};


// ==================================================================
// OrgStructure
// ==================================================================
    [Association, Aggregation, Version ("2.6.0"), Description (
    "CIM_OrgStructure is an association used to establish "
    "parent-child relationships between OrganizationalEntity "
    "instances. This is used to capture organizational relationships "
    "between object instances such as those that are imported from an "
    "LDAP-accessible directory.") ]
class CIM_OrgStructure {

    [Key, Max (1), Aggregate,
        Description ("The organizational parent in this association.")]
    CIM_OrganizationalEntity REF Parent;

    [Key, Description (
        "The organizational child in this association, "
        "i.e., the sub-unit or other owned object instance.") ]
    CIM_OrganizationalEntity REF Child;
};
```

```
// ===================================================================
// Organization
// ===================================================================
[Version ("2.6.0"), Description (
    "The Organization class is used to represent an organization such "
    "as a corporation or other autonomous entity. The class is "
    "defined so as to incorporate commonly-used LDAP attributes to "
    "permit implementations to easily derive this information from "
    "LDAP-accessible directories. This class's properties are a "
    "subset of a related class, OtherOrganizationInformation, which "
    "defines all the group properties in array form for "
    "directory compatibility.") ]
class CIM_Organization : CIM_OrganizationalEntity {

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Key, MaxLen (1024),Description (
        "The Name property defines the label by which the object is "
        "known. In the case of an LDAP-derived instance, the Name "
        "property value may be set to the distinguishedName of the "
        "LDAP-accessed object instance.") ]
    string Name;

    [MaxLen (128), Description (
        "This property describes the kind of business performed by an "
        "organization.") ]
    string BusinessCategory;

    [Description (
        "The organization's facsimile telephone number.") ]
    string FacsimileTelephoneNumber;

    [Description (
        "This property contains the name of a locality, such as a "
        "city, county, or other geographic region.") ]
    string LocalityName;

    [Description (
        "Based on RFC 1274, the mail box addresses for the "
        "organization as defined in RFC 822.") ]
    string Mail;

    [Required, Description (
        "The name of the organization.") ]
    string OrganizationName;

    [Description (
        "The Postal Address property values specify the address "
        "information required for the physical delivery of postal "
        "messages by the postal authority to the organization.") ]
    string PostalAddress[];

    [MaxLen (40), Description (
        "The Postal Code property specifies the postal code of the "
        "organization. If this value is present it will be part of "
        "the object's postal address.") ]
    string PostalCode;

    [Description (
        "The State or Province Name property specifies a state or "
        "province.") ]
    string StateOrProvince;

    [MaxLen (32), Description (
        "The Telephone Number property specifies a telephone number "
        "of the organization; e.g., + 44 582 10101).") ]
    string TelephoneNumber;
};

// ==================================================================
// OrgUnit
// ==================================================================
[Version ("2.6.0"), Description (
    "The OrgUnit class is used to represent a sub-unit of an "
    "organization such as a division or department. The class is "
    "defined so as to incorporate commonly used LDAP attributes to "
    "permit implementations to easily derive this information from "
    "LDAP-accessible directories. This class's properties are a "
    "subset of a related class, OtherOrgUnitInformation, which "
    "defines all the group properties and in array form for "
    "directory compatibility.") ]
class CIM_OrgUnit : CIM_OrganizationalEntity {

    [Key, MaxLen (256), Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to "
        "be uniquely identified.") ]
    string CreationClassName;

    [Key, MaxLen (1024),Description (
        "The Name property defines the label by which the object is "
        "known. In the case of an LDAP-derived instance, the Name "
        "property value may be set to the distinguishedName of the "
        "LDAP-accessed object instance.") ]
    string Name;

    [MaxLen (128), Description (
        "This property describes the kind of business performed by an "
        "organizational unit.") ]
    string BusinessCategory;
    [Description (
        "The organizational unit's facsimile telephone number.") ]
    string FacsimileTelephoneNumber;

    [Description (
        "This property contains the name of a locality, such as a "
        "city, county, or other geographic region.") ]
    string LocalityName;

    [Required, Description (
        "The name of the organizational unit.") ]
    string OU;

    [Description (
        "The Postal Address property values specify the address "
        "information required for the physical delivery of postal "
        "messages by the postal authority to the organizational "
        "unit.") ]
    string PostalAddress[];

    [MaxLen (40), Description (
        "The Postal Code property specifies the postal code of the "
        "organizational unit. If this value is present it will be "
        "part of the object's postal address.") ]
    string PostalCode;

    [Description (
        "The State or Province Name property specifies a state or "
        "province.") ]
    string StateOrProvince;

    [MaxLen (32), Description (
        "The Telephone Number property specifies a telephone number "
        "of the organizational unit; e.g., + 44 582 10101).") ]
    string TelephoneNumber;
};

// ==================================================================
// UserEntity
// ==================================================================
[Abstract, Version ("2.6.0"), Description (
    "UserEntity is an abstract class that represents users.") ]
class CIM_UserEntity : CIM_OrganizationalEntity {
};

// ==================================================================
// UserContact
// ==================================================================
[Experimental, Version ("2.7.1000"), Description (
    "UserContact contains the details related to contacting a "
    "user. This information will be promoted to UserEntity in a "
    "later major release of CIM, but cannot be promoted at this "
    "time since UserEntity is abstract today, with no keys. Key "
    "and required properties cannot be added to a class in a minor "
    "version release since this breaks backward compatibility.") ]
class CIM_UserContact : CIM_UserEntity {

    [Key, Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to be "
        "uniquely identified."),
    MaxLen ( 256) ]
    string CreationClassName;

    [Key, Description (
        "The Name property defines the label by which the object is "
        "known. In the case of an LDAP-derived instance, the Name "
        "property value may be set to the distinguished name of the "
        "LDAP-accessed object instance."),
    MaxLen ( 1024) ]
    string Name;

    [Description (
        "The Given Name property is used for the part of a person's "
        "name that is not their surname nor their middle name (i.e., "
        "their first name).") ]
    string GivenName;

    [Required, Description (
        "The Surname property specifies the linguistic construct "
        "that normally is inherited by an individual from the "
        "individual's parent or assumed by marriage, and by which "
        "the individual is commonly known.") ]
    string Surname;

    [Description (
        "Based on RFC 1274, the mail box addresses for the person as "
        "defined in RFC 822.") ]
    string Mail;

    [Description (
        "A User ID property. Based on RFC 1274, the property may be "
        "used to specify a computer system login name.") ]
    string UserID;

    [Description (
        "This property contains the name of a locality, such as a "
        "city, county, or other geographic region.") ]
    string LocalityName;

    [Description (
        "The Postal Address property values specify the address "
        "information required for the physical delivery of postal "
        "messages by the postal authority to the person.") ]
    string PostalAddress[];
    [Description (
        "The State or Province Name property specifies a state or "
        "province.") ]
    string StateOrProvince;

    [Description (
        "The Postal Code property specifies the postal code of the "
        "organization. If this value is present it will be part of "
        "the object's postal address."),
    MaxLen ( 40) ]
    string PostalCode;

    [Description (
        "The Telephone Number property specifies a telephone number "
        "of the organization; e.g., + 44 582 10101)."),
    MaxLen ( 32) ]
    string TelephoneNumber;
};

// ===================================================================
// Identity
// ===================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "An instance of an Identity represents a ManagedElement that "
        "acts as a security principal within the scope in which it is "
        "defined and authenticated. ManagedElements with Identities "
        "can be OrganizationalEntities, Services, Systems, etc. The "
        "ManagedElement 'behind' an Identity is described using the "
        "AssignedIdentity association. \n"
        "\n"
        "Within a given security context, an Identity may be imparted "
        "a level of trust, usually based on its credentials. Trust "
        "levels are indicated using a related CIM_SecuritySensitivity "
        "instance. Whether an Identity is currently authenticated is "
        "evaluated by checking the CurrentlyAuthenticated boolean "
        "property. This property is set and cleared by the security "
        "infrastructure, and should only be readable within the "
        "management infrastructure. The conditions that must be "
        "met/authenticated in order for an Identity's "
        "CurrentlyAuthenticated Boolean to be TRUE are defined using "
        "a subclass of PolicyCondition - AuthenticationCondition. The "
        "inheritance tree for AuthenticationCondition is defined in "
        "the CIM Policy Model. \n"
        "\n"
        "Subclasses of Identity may include specific information "
        "related to a given AuthenticationService or authority (such "
        "as a security token or computer hardware port/communication "
        "details) that more specifically determine the authenticity "
        "of the Identity. An instance of Identity may be persisted "
        "even though it is not CurrentlyAuthenticated, in order to "
        "maintain static relationships to Roles, associations to "
        "accounting information, and policy data defining "
        "authentication requirements. Note, however, that when an "
        "Identity is not authenticated "
        "(CurrentlyAuthenticated = FALSE), then Privileges or rights"
        "SHOULD NOT be authorized. The lifetime, validity, and "
        "propagation of the Identity is dependent on a security "
        "infrastructure's policies.") ]
class CIM_Identity : CIM_ManagedElement {

    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this "
        "class. In order to ensure uniqueness within the NameSpace, "
        "the value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "    <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked, "
        "or otherwise unique name that is owned by the business "
        "entity creating/defining the InstanceID, or is a registered "
        "ID that is assigned to the business entity by a recognized "
        "global authority (This is similar to the <Schema "
        "Name>_<Class Name> structure of Schema class names.) In "
        "addition, to ensure uniqueness <OrgID> MUST NOT contain a "
        "colon (':'). When using this algorithm, the first colon to "
        "appear in InstanceID MUST appear between <OrgID> and "
        "<LocalID>. \n"
        "<LocalID> is chosen by the business entity and SHOULD not "
        "be reused to identify different underlying (real-world) "
        "elements. If the above 'preferred' algorithm is not used, "
        "the defining entity MUST assure that the resultant "
        "InstanceID is not reused across any InstanceIDs produced "
        "by this or other providers for this instance's NameSpace. \n"
        "For DMTF-defined instances, the 'preferred' algorithm "
        "MUST be used with the <OrgID> set to 'CIM'.") ]
    string InstanceID;

    [Description (
        "Boolean indicating whether this Identity has been "
        "authenticated, and is currently known within the scope of "
        "an AuthenticationService or authority. By default, "
        "authenticity SHOULD NOT be assumed. This property is set "
        "and cleared by the security infrastructure, and should only "
        "be readable within the management infrastructure. Note "
        "that its value, alone, may not be sufficient to determine "
        "authentication/authorization, in that properties of an "
        "Identity subclass (such as a security token or computer "
        "hardware port/communication details) may be required by "
        "the security infrastructure.") ]
    boolean CurrentlyAuthenticated = FALSE;
};

// ==================================================================
// AssignedIdentity
// ==================================================================
    [Association, Experimental, Version ("2.7.1000"), Description (
        "This relationship associates an Identity to a specific "
        "ManagedElement, whose trust is represented.") ]
class CIM_AssignedIdentity {

    [Key, Description (
        "An Identity of the referenced ManagedElement.") ]
    CIM_Identity REF IdentityInfo;

    [Key, Max ( 1), Description (
        "The ManagedElement assigned to a specific Identity.") ]
    CIM_ManagedElement REF ManagedElement;
};

// ==================================================================
// Credential
// ==================================================================
[Abstract, Version ("2.7.0"), Description (
    "Subclasses of CIM_Credential define materials, "
    "information, or other data which are used to prove the "
    "identity of an entity. Generally, there may be some shared "
    "information, or credential material which is used to "
    "identify and authenticate ones self in the process of "
    "gaining access to, or permission to use, an Account. "
    "Such credential material may be used to authenticate a "
    "user’s access identity initially, and additionally on "
    "an ongoing basis during the course of a connection or "
    "other security association, as proof that each received "
    "message or communication came from the owning user access of "
    "that credential material.") ]
class CIM_Credential : CIM_ManagedElement {

    [Description (
        "The date and time when the credential was issued. Use a "
        "value of all 0s if this information is not applicable.") ]
    datetime Issued;

    [Description (
        "The date and time when the credential expires (and is "
        "not appropriate for use for authentication/"
        "authorization). Use a value of all 9s if this information "
        "is not applicable. Note that this property does not define "
        "how the expiration is set - but that there IS an expiration. "
        "The property may be set to either a specific date/time or "
        "an interval (calculated from the Issued datetime). For "
        "example, for a Certificate Authority-signed public key, the "
        "expiration is determined by the CA. Another example is a "
        "voice mail password that expires 60 days after it is "
        "set/issued.") ]
    datetime Expires;
};
```

```
// ==================================================================
// BoardingPass
// ==================================================================
[Experimental, Version ("2.8.0"), Description (
    "A kind of CIM_Credential identifying the specific boarding pass "
    "associated with an Identity (via the CredentialForIdentity "
    "relationship).") ]
class MaD_BoardingPass : CIM_Credential {

    [Key, Description (
        "InstanceID opaquely identifies a unique instance of "
        "BoardingPass. The InstanceID must be unique within a "
        "namespace. In order to ensure uniqueness, the value of "
        "InstanceID SHOULD be constructed in the following manner: \n"
        "    <Vendor ID>:<ID> \n"
        "<Vendor ID> MUST include a copyrighted, trademarked, or "
        " otherwise unique name that is owned by the business entity "
        "or a registered ID that is assigned to the business entity "
        "that is defining the InstanceID. (This is similar to the "
        "<Schema Name>_<Class Name> structure of Schema class names.) "
        "The purpose of <Vendor ID> is to ensure that <ID> is truly "
        "unique across multiple vendor implementations. If such a "
        "name is not used, the defining entity MUST assure that the "
        "<ID> portion of the Instance ID is unique when compared with "
        "other instance providers. For DMTF-defined instances, the "
        "<Vendor ID> is 'CIM'. \n"
        "<ID> SHOULD include the PassID string data.") ]
    string InstanceID;

    [Description (
        "The identifying information for this instance of the "
        "BoardingPass class.") ]
    string PassID;
};

// ==================================================================
// SecuritySensitivity
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "Defines a security sensitivity level, which can be "
        "associated with a ManagedElement. A simple example is to "
        "define security levels representing 'low sensitivity', "
        "'moderately sensitive', 'high sensitivity', and "
        "'national security' classification levels. An element is"
        "assigned a security level via the association, "
        "ElementSecurityLevel. It can be assigned to any "
        "ManagedElement, such as Locations, Identities, Roles, "
        "Systems, Services, and LogicalFiles. \n"
        "A security level is determined by a business, organization, "
        "and/or government based on the need to protect data and "
        "entities from attack, loss, abuse, or unauthorized, "
        "disclosure, and the ramifications if this protection "
        "is not maintained.") ]
class CIM_SecuritySensitivity : CIM_ManagedElement {
    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this "
        "class. In order to ensure uniqueness within the NameSpace, "
        "the value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "    <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked, "
        "or otherwise unique name that is owned by the business "
        "entity creating/defining the InstanceID, or is a registered "
        "ID that is assigned to the business entity by a recognized "
        "global authority (This is similar to the <Schema "
        "Name>_<Class Name> structure of Schema class names.) In "
        "addition, to ensure uniqueness <OrgID> MUST NOT contain a "
        "colon (':'). When using this algorithm, the first colon to "
        "appear in InstanceID MUST appear between <OrgID> and "
        "<LocalID>. \n"
        "<LocalID> is chosen by the business entity and SHOULD not "
        "be reused to identify different underlying (real-world) "
        "elements. If the above 'preferred' algorithm is not used, "
        "the defining entity MUST assure that the resultant "
        "InstanceID is not reused across any InstanceIDs produced "
        "by this or other providers for this instance's NameSpace. \n"
        "For DMTF-defined instances, the 'preferred' algorithm "
        "MUST be used with the <OrgID> set to 'CIM'.") ]
    string InstanceID;

    [Description (
        "A string defining the security sensitivity level.") ]
    string SecurityLevel;
};

// ===================================================================
// ElementSecuritySensitivity
// ===================================================================
    [Association, Experimental, Version ("2.7.1000"), Description (
        "This association relates a security sensitivity level to a "
        "ManagedElement. It describes the level assigned to the "
        "element.") ]
class CIM_ElementSecuritySensitivity {

    [Key, Description (
        "The security sensitivity level of the element.") ]
    CIM_SecuritySensitivity REF SecurityLevel;

    [Key, Description (
        "The element which is assigned a sensitivity level.") ]
    CIM_ManagedElement REF ManagedElement;
};
// ====================================================================
// end of file
// ====================================================================
```

A.9 Policy Information

```
// ====================================================================
// Title:       Mobile and Directory Schema - Subset of CIM Policy
//              Version 2.8 Model to support general policies and
//              Identity authentication rules
// Filename:    MaD_Policy11.mof
// Version:     1.1.0
// Status:      Preliminary
// Date:        11/10/03 (Initial release)
// ====================================================================
// Copyright 2003 The Open Group and Distributed Management Task
// Force, Inc. (DMTF). All rights reserved.
// ====================================================================

#pragma Locale ("en_US")

// ===================================================================
// Policy
// ===================================================================
    [Abstract, Version ("2.6.0"), Description (
        "An abstract class defining the common properties of the "
       "policy managed elements derived from CIM_Policy. The "
       "subclasses are used to create rules and groups of rules that "
       "work together to form a coherent set of policies within an "
       "administrative domain or set of domains.") ]
class CIM_Policy : CIM_ManagedElement {

    [Description (
        "A user-friendly name of this policy-related object.") ]
    string CommonName;

    [Description (
        "An array of keywords for characterizing/categorizing "
        "policy objects. Keywords are of one of two types: \n"
        "(1) Keywords defined in this and other MOFs, or in DMTF "
        "white papers. These keywords provide a vendor-independent, "
        "installation-independent way of characterizing policy "
        "objects. \n"
        "(2) Installation-dependent keywords for characterizing "
        "policy objects. Examples include 'Engineering', 'Billing', "
        "and 'Review in December 2000'. \n"
        "This MOF defines the following keywords: 'UNKNOWN', "
        "'CONFIGURATION', 'USAGE', 'SECURITY', 'SERVICE', "
        "'MOTIVATIONAL', 'INSTALLATION', and 'EVENT'. These "
        "concepts are self-explanatory and are further discussed in "
        "the SLA/Policy White Paper. One additional keyword is "
        "defined: 'POLICY'. The role of this keyword is to identify "
        "policy-related instances that may not be otherwise "
        "identifiable, in some implementations. The keyword "
        "'POLICY' is NOT mutually exclusive of the other keywords "
        "specified above.") ]
    string PolicyKeywords[];
};
```

```
// ===================================================================
// PolicySet
// ===================================================================
    [Abstract, Version ("2.7.1000"), Description (
        "PolicySet is an abstract class that represents a set of "
        "policies that form a coherent set. The set of contained "
        "policies has a common decision strategy and a common set of "
        "policy roles (defined via the PolicySetInRole Collection "
        "association). Subclasses include PolicyGroup and "
        "PolicyRule.") ]
class CIM_PolicySet : CIM_Policy {

    [Description (
        "PolicyDecisionStrategy defines the evaluation method used "
        "for policies contained in the PolicySet. FirstMatching "
        "enforces the actions of the first rule that evaluates to "
        "TRUE. It is the only value currently defined."),
    ValueMap { "1"},
    Values { "First Matching"} ]
    uint16 PolicyDecisionStrategy;

    [Description (
        "Indicates whether this PolicySet is administratively "
        "enabled, administratively disabled, or enabled for debug. "
        "The \"EnabledForDebug\" property value is deprecated and, "
        "when it or any value not understood by the receiver is "
        "specified, the receiving enforcement point treats the "
        "PolicySet as \"Disabled\". To determine if a PolicySet is "
        "\"Enabled\", the containment hierarchy specified by the "
        "PolicySetComponent aggregation is examined and the Enabled "
        "property values of the hierarchy are AND’ed together. Thus, "
        "for example, everything aggregated by a PolicyGroup may be "
        "disabled by setting the Enabled property in the PolicyGroup "
        "instance to \"Disabled\" without changing the Enabled "
        "property values of any of the aggregated instances. The "
        "default value is 1 (\"Enabled\")."),
    ValueMap { "1", "2", "3"},
    Values { "Enabled", "Disabled", "Enabled For Debug"} ]
    uint16 Enabled = 1;
};

// ===================================================================
// PolicyRule
// ===================================================================
    [Version ("2.7.0"), Description (
        "The central class used for representing the 'If Condition "
        "then Action' semantics of a policy rule. A PolicyRule "
        "condition, in the most general sense, is represented as "
        "either an Ored set of AND’ed conditions (Disjunctive Normal "
        "Form, or DNF) or an AND’ed set of OR’ed conditions "
        "(Conjunctive Normal Form, or CNF). Individual conditions"
        "may either be negated (NOT C) or unnegated (C) "
        "The actions specified by a PolicyRule are to be performed "
```

```
"if and only if the PolicyRule condition (whether "
"it is represented in DNF or CNF) evaluates to TRUE. \n"
"The conditions and actions associated with a PolicyRule are "
"modeled, respectively, with subclasses of PolicyCondition "
"and PolicyAction. These condition and action objects are "
"tied to instances of PolicyRule by the "
"PolicyConditionInPolicyRule and "
"PolicyActionInPolicyRule aggregations. \n"
"A PolicyRule may also be associated with one or more policy "
"time periods, indicating the schedule according to which the "
"policy rule is active and inactive.  In this case it is the "
"PolicySetValidityPeriod aggregation that provides this "
"linkage. \n"
"The PolicyRule class uses the property ConditionListType to "
"indicate whether the conditions for the rule are in DNF "
"(disjunctive normal form), CNF (conjunctive normal form) or, "
"in the case of a rule with no conditions, as an "
"UnconditionalRule. The PolicyConditionInPolicyRule "
"aggregation contains two additional properties to complete "
"the representation of the Rule's conditional expression. The "
"first of these properties is an integer to partition the "
"referenced PolicyConditions into one or more groups, and the "
"second is a Boolean to indicate whether a referenced "
"Condition is negated. An example shows how ConditionListType "
"and these two additional properties provide a unique "
"representation of a set of PolicyConditions in either "
"DNF or CNF. \n"
"Suppose we have a PolicyRule that aggregates five "
"PolicyConditions C1 through C5, with the following values in "
"the properties of the five PolicyConditionInPolicyRule "
"associations: \n"
"C1: GroupNumber = 1, ConditionNegated = FALSE \n"
"C2: GroupNumber = 1, ConditionNegated = TRUE \n"
"C3: GroupNumber = 1, ConditionNegated = FALSE \n"
"C4: GroupNumber = 2, ConditionNegated = FALSE \n"
"C5: GroupNumber = 2, ConditionNegated = FALSE \n"
"If ConditionListType = DNF, then the overall condition "
"for the PolicyRule is: \n"
"(C1 AND (NOT C2) AND C3) OR (C4 AND C5) \n"
"On the other hand, if ConditionListType = CNF, then the "
"overall condition for the PolicyRule is: \n"
"(C1 OR (NOT C2) OR C3) AND (C4 OR C5) \n"
"In both cases, there is an unambiguous specification of the "
"overall condition that is tested to determine whether to "
"perform the PolicyActions associated with the PolicyRule. \n"
"PolicyRule instances may also be used to aggregate other "
"PolicyRules and/or PolicyGroups. When used in this way to "
"implement nested rules, the conditions of the aggregating "
"rule apply to the subordinate rules as well. However, any "
"side-effects of condition evaluation or the execution of "
"actions MUST NOT affect the result of the evaluation of "
"other conditions evaluated by the rule engine in the same "
"evaluation pass. That is, an implementation of a rule engine "
"MAY evaluate all conditions in any order before applying the "
```

```
        "priority and determining which actions are to be executed.") ]
class CIM_PolicyRule : CIM_PolicySet {

    [Key, Propagated ( "CIM_System.CreationClassName"), Description (
        "The scoping System's CreationClassName."),
    MaxLen ( 256) ]
    string SystemCreationClassName;

    [Key, Propagated ( "CIM_System.Name"), Description (
        "The scoping System's Name."),
    MaxLen ( 256) ]
    string SystemName;

    [Key, Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to be "
        "uniquely identified."),
    MaxLen ( 256) ]
    string CreationClassName;

    [Key, Description (
        "A user-friendly name of this PolicyRule."),
    MaxLen ( 256) ]
    string PolicyRuleName;

    [Description (
        "Indicates whether the list of PolicyConditions associated "
        "with this PolicyRule is in disjunctive normal form (DNF), "
        "conjunctive normal form (CNF), or has no conditions (i.e., "
        "is an UnconditionalRule) and is automatically evaluated to "
        "\"True.\" The default value is 1 (\"DNF\")."),
    ValueMap { "0", "1", "2"},
    Values { "Unconditional Rule", "DNF", "CNF"} ]
    uint16 ConditionListType = 1;

    [Description (
        "A free-form string that can be used to provide guidelines "
        "on how this PolicyRule should be used.") ]
    string RuleUsage;

    [Description (
        "This property gives a policy administrator a way of "
        "specifying how the ordering of the PolicyActions associated "
        "with this PolicyRule is to be interpreted. Three values "
        "are supported: \n"
        "- mandatory(1): Do the actions in the indicated order, or "
        "don't do them at all. \n"
        "- recommended(2): Do the actions in the indicated order if "
        "you can, but if you can't do them in this order, do them in "
        "another order if you can. \n"
        "- dontCare(3): Do them - I don't care about the order. \n"
        "The default value is 3 (\"DontCare\")."),
    ValueMap { "1", "2", "3"},
    Values { "Mandatory", "Recommended", "Dont Care"} ]
    uint16 SequencedActions = 3;

    [Description (
        "ExecutionStrategy defines the strategy to be used in "
        "executing the sequenced actions aggregated by this "
        "PolicyRule. There are three execution strategies: \n"
        "Do Until Success - execute actions according to predefined "
        "order, until successful execution of a single action. \n"
        "Do All - execute ALL actions which are part of the modeled "
        "set, according to their predefined order. Continue doing "
        "this, even if one or more of the actions fails. \n"
        "Do Until Failure - execute actions according to predefined "
        "order, until the first failure in execution of an action "
        "instance."),
    ValueMap { "1", "2", "3"},
    Values { "Do Until Success", "Do All", "Do Until Failure"} ]
    uint16 ExecutionStrategy;
};

// ==================================================================
// AuthenticationRule
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class representing a company's and/or administrator's "
        "authentication requirements for a CIM_Identity. The "
        "PolicyConditions collected by an instance of "
        "AuthenticationRule describe the various requirements under "
        "which a CIM_Identity's CurrentlyAuthenticated Boolean is set "
        "to TRUE. Note that the CIM_Identities which are "
       "authenticated are tied to the Rule by the association, "
       "PolicySet AppliesToElement. \n"
       "At this time, there are no actions associated with this "
       "PolicyRule. This is because the actions are implicit. When "
       "the conditions of the rule are met, then the "
       "CurrentlyAuthenticated Boolean properties of the associated "
       "instances of CIM_Identity are set to TRUE.") ]
class CIM_AuthenticationRule : CIM_PolicyRule {
};

// ==================================================================
// PolicyCondition
// ==================================================================
    [Abstract, Version ("2.6.0"), Description (
        "A class representing a rule-specific or reusable policy "
        "condition to be evaluated in conjunction with a Policy Rule. "
        "Since all operational details of a PolicyCondition are "
        "provided in subclasses of this object, this class is "
        "abstract.") ]
class CIM_PolicyCondition : CIM_Policy {

    [Key, Description (
        "The name of the class or the subclass used in the creation "
        "of the System object in whose scope this PolicyCondition is "
        "defined. \n"
        "This property helps to identify the System object in whose "
        "scope this instance of PolicyCondition exists. For a "
        "rule-specific PolicyCondition, this is the System in whose "
        "context the PolicyRule is defined. For a reusable "
        "PolicyCondition, this is the instance of PolicyRepository "
        "(which is a subclass of System) that holds the Condition. \n"
        "Note that this property, and the analogous property "
        "SystemName, do not represent propagated keys from an "
        "instance of the class System. Instead, they are properties "
        "defined in the context of this class, which repeat the "
        "values from the instance of System to which this "
        "PolicyCondition is related, either directly via the "
        "PolicyConditionInPolicyRepository association or indirectly "
        "via the PolicyConditionInPolicyRule aggregation."),
    MaxLen ( 256) ]
    string SystemCreationClassName;

    [Key, Description (
        "The name of the System object in whose scope this "
        "PolicyCondition is defined. \n"
        "This property completes the identification of the System "
        "object in whose scope this instance of PolicyCondition "
        "exists. For a rule-specific PolicyCondition, this is the "
        "System in whose context the PolicyRule is defined. For a "
        "reusable PolicyCondition, this is the instance of "
        "PolicyRepository (which is a subclass of System) that holds "
        "the Condition."),
    MaxLen ( 256) ]
    string SystemName;

    [Key, Description (
        "For a rule-specific PolicyCondition, the CreationClassName "
        "of the PolicyRule object with which this Condition is "
        "associated. For a reusable Policy Condition, a special "
        "value, 'NO RULE', should be used to indicate that this "
        "Condition is reusable and not associated with a single "
        "PolicyRule."),
    MaxLen ( 256) ]
    string PolicyRuleCreationClassName;

    [Key, Description (
        "For a rule-specific PolicyCondition, the name of the "
        "PolicyRule object with which this Condition is associated. "
        "For a reusable PolicyCondition, a special value, 'NO RULE', "
        "should be used to indicate that this Condition is reusable "
        "and not associated with a single PolicyRule."),
    MaxLen ( 256) ]
    string PolicyRuleName;

    [Key, Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to be "
        "uniquely identified."),
    MaxLen ( 256) ]
    string CreationClassName;

    [Key, Description (
        "A user-friendly name of this PolicyCondition."),
    MaxLen ( 256) ]
    string PolicyConditionName;
};

// ==================================================================
// PolicyTimePeriodCondition
// ==================================================================
    [Version ("2.7.0"), Description (
        "This class provides a means of representing the time periods "
        "during which a PolicySet is valid; i.e., active. At all "
        "times that fall outside these time periods, the PolicySet "
        "has no effect. A PolicySet is treated as valid at ALL times, "
        "if it does not specify a PolicyTimePeriodCondition. \n"
        "In some cases a Policy Consumer may need to perform certain "
        "setup/cleanup actions when a PolicySet becomes active/ "
        "inactive. For example, sessions that were established while "
        "a PolicySet was active might need to be taken down when the "
        "PolicySet becomes inactive. In other cases, however, such "
        "sessions might be left up. In this case, the effect of "
        "deactivating the PolicySet would just be to prevent the "
        "establishment of new sessions. \n"
        "Setup/cleanup behaviors on validity period transitions are "
        "not currently addressed by the Policy Model, and must be "
        "specified in 'guideline' documents or via subclasses of "
        "CIM_PolicySet, CIM_PolicyTimePeriod Condition, or other "
        "concrete subclasses of CIM_Policy. If such behaviors need to "
        "be under the control of the policy administrator, then a "
        "mechanism to allow this control must also be specified in "
        "the subclasses. \n"
        "PolicyTimePeriodCondition is defined as a subclass of "
        "PolicyCondition. This is to allow the inclusion of "
        "time-based criteria in the AND/OR condition definitions "
        "for a PolicyRule. \n"
        "Instances of this class may have up to five properties "
        "identifying time periods at different levels. The values of "
        "all the properties present in an instance are AND’ed "
        "together to determine the validity period(s) for the "
        "instance. For example, an instance with an overall validity "
        "range of January 1, 2000 through December 31, 2000; a month "
        "mask that selects March and April; a day-of-the-week mask "
        "that selects Fridays; and a time of day range of 0800 "
        "through 1600 would be represented using the following time "
        "periods: \n"
        "Friday, March 5, 2000, from 0800 through 1600; \n"
        "Friday, March 12, 2000, from 0800 through 1600; \n"
        "Friday, March 19, 2000, from 0800 through 1600; \n"
```

```
            "Friday, March 26, 2000, from 0800 through 1600; \n"
            "Friday, April 2, 2000, from 0800 through 1600; \n"
            "Friday, April 9, 2000, from 0800 through 1600; \n"
            "Friday, April 16, 2000, from 0800 through 1600; \n"
            "Friday, April 23, 2000, from 0800 through 1600; \n"
            "Friday, April 30, 2000, from 0800 through 1600. \n"
            "Properties not present in an instance of "
            "PolicyTimePeriodCondition are implicitly treated as having "
            "their value 'always enabled'. Thus, in the example above, "
            "the day-of-the-month mask is not present, and so the "
            "validity period for the instance implicitly includes a "
            "day-of-the-month mask that selects all days of the month. If "
            "this 'missing property' rule is applied to its fullest, we "
            "see that there is a second way to indicate that a PolicySet "
            "is always enabled: associate with it an instance of "
            "PolicyTimePeriodCondition whose only properties with"
            "specific values are its key properties.") ]
    class CIM_PolicyTimePeriodCondition : CIM_PolicyCondition {

        [Description (
            "This property identifies an overall range of calendar dates "
            "and times over which a PolicySet is valid. It is formatted "
            "as a string representing a start date and time, in which "
            "the character 'T' indicates the beginning of the time "
            "portion, followed by the solidus character '/', followed by "
            "a similar string representing an end date and time. The "
            "first date indicates the beginning of the range, while the "
            "second date indicates the end. Thus, the second date and "
            "time must be later than the first. Date/times are "
            "expressed as substrings of the form yyyymmddThhmmss. For "
            "example: \n"
            "    20000101T080000/20000131T120000 \n"
            "defines January 1, 2000, 0800 through January 31, 2000, "
            "noon. \n"
            "There are also two special cases in which one of the "
            "date/time strings is replaced with a special string defined "
            "in RFC 2445. \n"
            "- If the first date/time is replaced with the string "
            "'THISANDPRIOR', then the property indicates that a "
            "PolicySet is valid [from now] until the date/time that "
            "appears after the '/'. \n"
            "- If the second date/time is replaced with the string "
            "'THISANDFUTURE', then the property indicates that a "
            "PolicySet becomes valid on the date/time that appears "
            "before the '/', and remains valid from that point on."),
        ModelCorrespondence {
            "CIM_PolicyTimePeriodCondition.MonthOfYearMask",
            "CIM_PolicyTimePeriodCondition.DayOfMonthMask",
            "CIM_PolicyTimePeriodCondition.DayOfWeekMask",
            "CIM_PolicyTimePeriodCondition.TimeOfDayMask",
            "CIM_PolicyTimePeriodCondition.LocalOrUtcTime"} ]
        string TimePeriod;

        [Description (
            "The purpose of this property is to refine the valid time "
            "period that is defined by the TimePeriod property, by "
            "explicitly specifying in which months the PolicySet is "
            "valid. These properties work together, with the TimePeriod "
            "used to specify the overall time period in which the "
            "PolicySet is valid, and the MonthOfYearMask used to pick "
            "out the months during which the PolicySet is valid. \n"
            "This property is formatted as an octet string, structured "
            "as follows: \n"
            "- A 4-octet length field, indicating the length of the "
            "entire octet string; this field is always set to 0x00000006 "
            "for this property; \n"
            "o A 2-octet field consisting of 12 bits identifying the 12 "
            "months of the year, beginning with January and ending with "
            "December, followed by 4 bits that are always set to '0'. "
            "For each month, the value '1' indicates that the policy is "
            "valid for that month, and the value '0' indicates that it "
            "is not valid. \n"
            "The value 0x000000060830, for example, indicates that a "
            "PolicySet is valid only in the months May, November, and "
            "December. \n"
            "If a value for this property is not provided, then the "
            "PolicySet is treated as valid for all twelve months, and "
            "only restricted by its TimePeriod property value and the "
            "other Mask properties."),
        OctetString,
        ModelCorrespondence {
            "CIM_PolicyTimePeriodCondition.TimePeriod",
            "CIM_PolicyTimePeriodCondition.LocalOrUtcTime"} ]
        uint8 MonthOfYearMask[];

        [Description (
            "The purpose of this property is to refine the valid time "
            "period that is defined by the TimePeriod property, by "
            "explicitly specifying in which days of the month the "
            "PolicySet is valid. These properties work together, with "
            "the TimePeriod used to specify the overall time period in "
            "which the PolicySet is valid, and the DayOfMonthMask used "
            "to pick out the days of the month during which the "
            "PolicySet is valid. \n"
            "This property is formatted as an octet string, structured "
            "as follows: \n"
            "- A 4-octet length field, indicating the length of the "
            "entire octet string; this field is always set to 0x0000000C "
            "for this property; \n"
            "- An 8-octet field consisting of 31 bits identifying the "
            "days of the month counting from the beginning, followed by "
            "31 more bits identifying the days of the month counting "
            "from the end, followed by 2 bits that are always set to "
            "'0'. For each day, the value '1' indicates that the "
            "PolicySet is valid for that day, and the value '0' "
            "indicates that it is not valid. \n"
            "The value 0x0000000C8000000100000000, for example, "
            "indicates that a PolicySet is valid on the first and last "
            "days of the month. \n"
            "For months with fewer than 31 days, the digits "
            "corresponding to days that the months do not have (counting "
            "in both directions) are ignored. \n"
            "If a value for this property is not provided, then the "
            "PolicySet is treated as valid for all days of the month, "
            "and only restricted by its TimePeriod property value and "
            "the other Mask properties."),
        OctetString,
            ModelCorrespondence {
            "CIM_PolicyTimePeriodCondition.TimePeriod",
            "CIM_PolicyTimePeriodCondition.LocalOrUtcTime"} ]
        uint8 DayOfMonthMask[];

        [Description (
            "The purpose of this property is to refine the valid time "
            "period that is defined by the TimePeriod property, by "
            "explicitly specifying in which days of the week the "
            "PolicySet is valid. These properties work together, with "
            "the TimePeriod used to specify the overall time period in "
            "which the PolicySet is valid, and the DayOfWeekMask used to "
            "pick out the days of the week during which the PolicySet is "
            "valid. \n"
            "This property is formatted as an octet string, structured "
            "as follows: \n"
            "- A 4-octet length field, indicating the length of the "
            "entire octet string; this field is always set to 0x00000005 "
            "for this property; \n"
            "- A 1-octet field consisting of 7 bits identifying the 7 "
            "days of the week, beginning with Sunday and ending with "
            "Saturday, followed by 1 bit that is always set to '0'. For "
            "each day of the week, the value '1' indicates that the "
            "PolicySet is valid for that day, and the value '0' "
            "indicates that it is not valid. \n"
            "The value 0x000000057C, for example, indicates that a "
            "PolicySet is valid Monday through Friday. \n"
            "If a value for this property is not provided, then the "
            "PolicySet is treated as valid for all days of the week, and "
            "only restricted by its TimePeriod property value and the "
            "other Mask properties."),
        OctetString,
        ModelCorrespondence {
            "CIM_PolicyTimePeriodCondition.TimePeriod",
            "CIM_PolicyTimePeriodCondition.LocalOrUtcTime"} ]
        uint8 DayOfWeekMask[];

        [Description (
            "The purpose of this property is to refine the valid time "
            "period that is defined by the TimePeriod property, by "
            "explicitly specifying a range of times in a day during "
            "which the PolicySet is valid. These properties work "
            "together, with the TimePeriod used to specify the overall "
            "time period in which the PolicySet is valid, and the "
            "TimeOfDayMask used to pick out the range of time periods in "
```

```
        "a given day of during which the PolicySet is valid. \n"
        "This property is formatted in the style of RFC 2445: a time "
        "string beginning with the character 'T', followed by the "
        "solidus character '/', followed by a second time string. "
        "The first time indicates the beginning of the range, while "
        "the second time indicates the end. Times are expressed as "
        "substrings of the form 'Thhmmss'. \n"
        "The second substring always identifies a later time than "
        "the first substring. To allow for ranges that span "
        "midnight, however, the value of the second string may be "
        "smaller than the value of the first substring. Thus, "
        "'T080000/T210000' identifies the range from 0800 until "
        "2100, while 'T210000/T080000' identifies the range from "
        "2100 until 0800 of the following day. \n"
        "When a range spans midnight, it by definition includes "
        "parts of two successive days. When one of these days is "
        "also selected by either the MonthOfYearMask, "
        "DayOfMonthMask, and/or DayOfWeekMask, but the other day is "
        "not, then the PolicySet is active only during the portion "
        "of the range that falls on the selected day. For example, "
        "if the range extends from 2100 until 0800, and the day of "
        "week mask selects Monday and Tuesday, then the PolicySet is "
        "active during the following three intervals: \n"
        "From midnight Sunday until 0800 Monday; \n"
        "From 2100 Monday until 0800 Tuesday; \n"
        "From 2100 Tuesday until 23:59:59 Tuesday. \n"
        "If a value for this property is not provided, then the "
        "PolicySet is treated as valid for all hours of the day, and "
        "only restricted by its TimePeriod property value and the "
        "other Mask properties."),
    ModelCorrespondence {
        "CIM_PolicyTimePeriodCondition.TimePeriod",
        "CIM_PolicyTimePeriodCondition.LocalOrUtcTime"} ]
    string TimeOfDayMask;

    [Description (
        "This property indicates whether the times represented in "
        "the TimePeriod property and in the various Mask properties "
        "represent local times or UTC times. There is no provision "
        "for mixing of local times and UTC times: the value of this "
        "property applies to all of the other time-related "
        "properties. TimePeriods are synchronized worldwide by "
        "using the enumeration value 'UTCTime'. If the goal is to "
        "synchronize worldwide on a particular local time (such as "
        "0300 - 0500 in New York), then if the TimePeriod property "
        "spans a Daylight Savings Time transition in New York, it "
        "will be necessary to create multiple instances of "
        "PolicyTimePeriodCondition, one based on the offset UTC-0500 "
        "for the part of each year when standard time is used in New "
        "York, and one based on the offset UTC-0400 for the part of "
        "each year when Daylight Savings Time is used there."),
    ValueMap { "1", "2"},
    Values { "Local Time", "UTC Time"},
    ModelCorrespondence {
        "CIM_PolicyTimePeriodCondition.TimePeriod",
        "CIM_PolicyTimePeriodCondition.MonthOfYearMask",
        "CIM_PolicyTimePeriodCondition.DayOfMonthMask",
        "CIM_PolicyTimePeriodCondition.DayOfWeekMask",
        "CIM_PolicyTimePeriodCondition.TimeOfDayMask"} ]
    uint16 LocalOrUtcTime;
};

// ==================================================================
// CompoundPolicyCondition
// ==================================================================
    [Version ("2.7.0"), Description (
        "CompoundPolicyCondition is used to represent compound "
        "conditions formed by aggregating simpler policy conditions. "
        "Compound conditions are constructed by associating "
        "subordinate condition terms together using the "
        "PolicyConditionInPolicyCondition aggregation.") ]
class CIM_CompoundPolicyCondition : CIM_PolicyCondition {

    [Description (
        "Indicates whether the list of CompoundPolicyConditions "
        "associated with this PolicyRule is in disjunctive normal "
        "form (DNF) or conjunctive normal form (CNF). The default "
        "value is 1 (\"DNF\")."),
    ValueMap { "1", "2"},
    Values { "DNF", "CNF" } ]
    uint16 ConditionListType;
};

// ==================================================================
// AuthenticationCondition
// ==================================================================
    [Experimental, Abstract, Version ("2.7.1000"), Description (
        "An abstract class whose subclasses describe one of a "
        "company’s and/or administrator's credential requirements, "
        "and/or other information that should be authenticated in "
        "order to establish/trust a CIM_Identity. The "
        "PolicyConditions collected by an instance of "
        "AuthenticationRule describe the various requirements under "
        "which a CIM_Identity’s CurrentlyAuthenticated Boolean is set "
        "to TRUE. Note that the CIM_Identities which are "
        "authenticated are specified through the AuthenticationRule, "
        "using the PolicySet AppliesToElement association.") ]
class CIM_AuthenticationCondition : CIM_PolicyCondition {
};

// ==================================================================
// SharedSecretAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "a specific identity whose shared secret should be "
        "authenticated.") ]
class CIM_SharedSecretAuthentication : CIM_AuthenticationCondition {

    [Description (
        "String defining the principal's ID whose secret is "
        "authenticated.") ]
    string IDOfPrincipal;

    [Description (
        "String defining a hostname, URI, or service/application "
        "name. It defines the specific system or service which "
        "provides the context for the shared secret.") ]
    string ContextOfSecret;
};

// ==================================================================
// AccountAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "a specific identity whose account credentials should be "
        "authenticated.") ]
class CIM_AccountAuthentication : CIM_AuthenticationCondition {

    [Description (
        "String defining the account's ID which is authenticated.") ]
    string AccountID;

    [Description (
        "String defining a hostname, URI, or other information "
        "identifying the system where the Account resides.") ]
    string AccountContext;
};

// ==================================================================
// BiometricAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "specific biometric data that should be authenticated.") ]
class CIM_BiometricAuthentication : CIM_AuthenticationCondition {

    [Description (
        "Integer enumeration identifying the biometric data that "
        "should be authenticated."),
    ValueMap { "1", "2", "3", "4", "5", "6","7", "8"},
    Values { "Other", "Facial", "Retina", "Mark", "Finger", "Voice",
        "DNA-RNA", "EEG"},
    ModelCorrespondence {
        "CIM_BiometricAuthentication.OtherBiometric"} ]
    uint16 TypeOfBiometric;

    [Description (
        "String specifying the biometric when the TypeOfBiometric "
        "property is set to 1, \"Other\"."),
    ModelCorrespondence {
        "CIM_BiometricAuthentication.TypeOfBiometric"} ]
    string OtherBiometric;
};

// ==================================================================
// PublicPrivateKeyAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "the specific public/private key pair that should be "
        "authenticated.") ]
class CIM_PublicPrivateKeyAuthentication : CIM_AuthenticationCondition{

    [Description (
        "Boolean indicating whether the key pair is self-issued "
        "(TRUE) or issued by a Certificate Authority (FALSE).") ]
    boolean SelfIssuedKey;

    [Description (
        "String holding the user's (distinguished) name.") ]
    string DistinguishedName;

    [Description (
        "String holding the public key data.") ]
    string PublicKey;
};

// ==================================================================
// KerberosAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "a user whose Kerberos ticket should be authenticated.") ]
class CIM_KerberosAuthentication : CIM_AuthenticationCondition {

    [Description (
        "String holding the user name for which the ticket is "
        "issued.") ]
    string UserName;
};
```

```
// ==================================================================
// DocumentAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "the specific document that should be authenticated.") ]
class CIM_DocumentAuthentication : CIM_AuthenticationCondition {

    [Description (
        "Integer enumeration identifying the document that should be "
        "authenticated."),
    ValueMap { "1", "2", "3", "4", "5", "6","7"},
    Values { "Other", "Passport", "Birth Certificate",
        "Credit Card", "Drivers License", "Membership Card",
        "Social Security Card"},
    ModelCorrespondence {
        "CIM_DocumentAuthentication.OtherDocument"} ]
    uint16 TypeOfDocument;

    [Description (
        "String specifying the document when the TypeOfDocument "
        "property is set to 1, \"Other\"."),
     ModelCorrespondence {
        "CIM_DocumentAuthentication.TypeOfDocument"} ]
    string OtherDocument;

    [Description (
        "Document identifier, such as a passport number.") ]
    string DocumentIdentifier;
};

// ==================================================================
// PhysicalCredentialAuthentication
// ==================================================================
    [Experimental, Version ("2.7.1000"), Description (
        "A class describing a company's and/or administrator's "
        "credential requirements that should be authenticated in "
        "order to establish/trust a CIM_Identity. This class defines "
        "the specific type of physical credential that should be "
        "authenticated.") ]
class CIM_PhysicalCredentialAuthentication:CIM_AuthenticationCondition{

    [Description (
        "Integer enumeration identifying the credential that should "
        "be authenticated."),
    ValueMap { "1", "2", "3", "4"},
    Values { "Other", "Magnetic Stripe Card", "Smart Card",
        "Password Generator Card"},
    ModelCorrespondence {
        "CIM_PhysicalCredentialAuthentication.OtherCredential"} ]
    uint16 TypeOfCredential;
    [Description (
        "String specifying the credential when the TypeOfCredential "
        "property is set to 1, \"Other\"."),
    ModelCorrespondence {
        "CIM_PhysicalCredentialAuthentication.TypeOfCredential"} ]
    string OtherCredential;

    [Description (
        "Identifier for the physical credential.") ]
    string PhysicalIdentifier;
};

// ==================================================================
// PolicyAction
// ==================================================================
    [Abstract, Version ("2.7.1000"), Description (
        "A class representing a rule-specific or reusable policy "
        "action to be performed if the PolicyConditions for a Policy "
        "Rule evaluate to TRUE. Since all operational details of a "
        "PolicyAction are provided in subclasses of this object, this "
        "class is abstract.") ]
class CIM_PolicyAction : CIM_Policy {

    [Key, Description (
        "The name of the class or the subclass used in the creation "
        "of the System object in whose scope this PolicyAction is "
        "defined. \n"
        "This property helps to identify the System object in whose "
        "scope this instance of PolicyAction exists. For a "
        "rule-specific PolicyAction, this is the System in whose "
        "context the PolicyRule is defined. For a reusable "
        "PolicyAction, this is the instance of PolicyRepository "
        "(which is a subclass of System) that holds the Action. \n"
        "Note that this property, and the analogous property "
        "SystemName, do not represent propagated keys from an "
        "instance of the class System. Instead, they are properties "
        "defined in the context of this class, which repeat the "
        "values from the instance of System to which this "
        "PolicyAction is related, either directly via the "
        "PolicyActionInPolicyRepository association or indirectly "
        "via the PolicyActionInPolicyRule aggregation."),
    MaxLen ( 256) ]
    string SystemCreationClassName;

    [Key, Description (
        "The name of the System object in whose scope this "
        "PolicyAction is defined. \n"
        "This property completes the identification of the System "
        "object in whose scope this instance of PolicyAction "
        "exists. For a rule-specific PolicyAction, this is the "
        "System in whose context the PolicyRule is defined. For a "
        "reusable PolicyAction, this is the instance of "
        "PolicyRepository (which is a subclass of System) that holds "
        "the Action."),
    MaxLen ( 256) ]
    string SystemName;

    [Key, Description (
        "For a rule-specific PolicyAction, the CreationClassName of "
        "the PolicyRule object with which this Action is "
        "associated. For a reusable PolicyAction, a special value, "
        "'NO RULE', should be used to indicate that this Action is "
        "reusable and not associated with a single PolicyRule."),
    MaxLen ( 256) ]
    string PolicyRuleCreationClassName;

    [Key, Description (
        "For a rule-specific PolicyAction, the name of the "
        "PolicyRule object with which this Action is associated. "
        "For a reusable PolicyAction, a special value, 'NO RULE', "
        "should be used to indicate that this Action is reusable and "
        "not associated with a single PolicyRule."),
    MaxLen ( 256) ]
    string PolicyRuleName;

    [Key, Description (
        "CreationClassName indicates the name of the class or the "
        "subclass used in the creation of an instance. When used "
        "with the other key properties of this class, this property "
        "allows all instances of this class and its subclasses to be "
        "uniquely identified."),
    MaxLen ( 256) ]
    string CreationClassName;

    [Key, Description (
        "A user-friendly name of this PolicyAction."),
    MaxLen ( 256) ]
    string PolicyActionName;

    [Experimental, Description (
        "DoActionLogging causes a log message to be generated when "
        "the action is performed.") ]
    boolean DoActionLogging;
};

// ===================================================================
// CompoundPolicyAction
// ===================================================================
    [Version ("2.6.0"), Description (
        "CompoundPolicyAction is used to represent an expression "
        "consisting of an ordered sequence of action terms. Each "
        "action term is represented as a subclass of the PolicyAction "
        "class. Compound actions are constructed by associating "
        "dependent action terms together using the "
        "PolicyActionInPolicyAction aggregation.") ]
class CIM_CompoundPolicyAction : CIM_PolicyAction {

    [Description (
        "This property gives a policy administrator a way of "
        "specifying how the ordering of the PolicyActions associated "
        "with this PolicyRule is to be interpreted. Three values "
        "are supported: \n"
        "- mandatory(1): Do the actions in the indicated order, or "
        "  don't do them at all. \n"
        "- recommended(2): Do the actions in the indicated order if "
        "  you can, but if you can't do them in this order, do "
        "  them in another order if you can. \n"
        "o dontCare(3): Do them -- I don't care about the order. \n"
        "The default value is 3 (\"DontCare\")."),
    ValueMap { "1", "2", "3"},
    Values { "Mandatory", "Recommended", "Dont Care"} ]
    uint16 SequencedActions;

    [Description (
        "ExecutionStrategy defines the strategy to be used in "
        "executing the sequenced actions aggregated by this "
        "CompoundPolicyAction. There are three execution "
        "strategies: \n"
        "- Do Until Success - Execute actions according to predefined "
        "  order, until successful execution of a single action. \n"
        "- Do All - Execute ALL actions which are part of the modeled "
        "  set, according to their predefined order. Continue doing "
        "  this, even if one or more of the actions fails. \n"
        "- Do Until Failure - Execute actions according to predefined "
        "  order, until the first failure in execution of an action "
        "  instance. \n"
        "The default value is 2 (\"Do All\")."),
    ValueMap { "1", "2", "3"},
    Values { "Do Until Success", "Do All", "Do Until Failure"} ]
    uint16 ExecutionStrategy;
};

// ==================================================================
// PolicyComponent
// ==================================================================
    [Association, Abstract, Aggregation, Version ("2.6.0"),
    Description (
        "CIM_PolicyComponent is a generic association used to "
        "establish 'part of' relationships between the subclasses of "
        "CIM_Policy. For example, the PolicyConditionInPolicyRule "
        "association defines that PolicyConditions are part of a "
        "PolicyRule.") ]
class CIM_PolicyComponent : CIM_Component {

    [Aggregate, Override ( "GroupComponent"), Description (
        "The parent Policy in the association.") ]
    CIM_Policy REF GroupComponent;

    [Override ( "PartComponent"), Description (
        "The child/part Policy in the association.") ]
    CIM_Policy REF PartComponent;
};
```

```
// ==================================================================
// PolicyInSystem
// ==================================================================
    [Association, Abstract, Version ("2.7.1000"), Description (
        "CIM_PolicyInSystem is a generic association used to "
        "establish dependency relationships between Policies and the "
        "Systems that host them. These Systems may be ComputerSystems "
        "where Policies are 'running' or they may be Policy "
        "Repositories where Policies are stored. This relationship is "
        "similar to the concept of CIM_Services being dependent on "
        "CIM_Systems as defined by the HostedService association. \n"
        "Cardinality is Max (1) for the Antecedent/System reference "
        "since Policies can only be hosted in at most one System "
        "context. Some subclasses of the association will further "
        "refine this definition to make the Policies Weak to Systems. "
        "Other subclasses of PolicyInSystem will define an optional "
        "hosting relationship. Examples of each of these are the "
        "PolicyRuleInSystem and PolicyConditionIn PolicyRepository "
        "associations, respectively.") ]
class CIM_PolicyInSystem : CIM_Dependency {

    [Override ( "Antecedent"), Max ( 1), Description (
        "The hosting System.") ]
    CIM_System REF Antecedent;

    [Override ( "Dependent"), Description (
        "The hosted Policy.") ]
    CIM_Policy REF Dependent;
};

// ==================================================================
// PolicySetComponent
// ==================================================================
    [Association, Aggregation, Version ("2.6.0"), Description (
        "PolicySetComponent is a concrete aggregation that collects "
        "instances of the subclasses of PolicySet (i.e., PolicyGroups "
        "and PolicyRules). Instances are collected in sets that use "
        "the same decision strategy. They are prioritized relative to "
        "each other, within the set, using the Priority property of "
        "this aggregation. \n"
        "Together, the PolicySet.PolicyDecisionStrategy and PolicySet "
        "Component.Priority properties determine the processing for "
        "the groups and rules contained in a PolicySet. A larger "
        "priority value represents a higher priority. Note that the "
        "Priority property MUST have a unique value when compared "
        "with others defined for the same aggregating PolicySet. "
        "Thus, the evaluation of rules within a set is "
        "deterministically specified.") ]
class CIM_PolicySetComponent : CIM_PolicyComponent {

    [Aggregate, Override ( "GroupComponent"), Description (
        "A PolicySet that aggregates other PolicySet instances.") ]
    CIM_PolicySet REF GroupComponent;
    [Override ( "PartComponent"), Description (
        "A PolicySet aggregated into a PolicySet.") ]
    CIM_PolicySet REF PartComponent;

    [Description (
        "A non-negative integer for prioritizing this PolicySet "
        "component relative to other elements of the same "
        "PolicySet. A larger value indicates a higher priority. "
        "The Priority property MUST have a unique value when "
        "compared with others defined for the same aggregating "
        "PolicySet.") ]
    uint16 Priority;
};

// ==================================================================
// PolicySetInSystem
// ==================================================================
    [Association, Abstract, Version ("2.6.0"), Description (
        "PolicySetInSystem is an abstract association class that "
        "represents a relationship between a System and a PolicySet "
        "used in the administrative scope of that system (e.g., "
        "AdminDomain, ComputerSystem). The Priority property is used "
        "to assign a relative priority to a PolicySet within the "
        "administrative scope in contexts where it is not a component "
        "of another PolicySet.") ]
class CIM_PolicySetInSystem : CIM_PolicyInSystem {

    [Override ( "Antecedent"), Min ( 1), Max ( 1), Description (
        "The System in whose scope a PolicySet is defined.") ]
    CIM_System REF Antecedent;

    [Override ( "Dependent"), Description (
        "A PolicySet named within the scope of a System.") ]
    CIM_PolicySet REF Dependent;

    [Description (
        "The Priority property is used to specify the relative "
        "priority of the referenced PolicySet when there are more "
        "than one PolicySet instances applied to a managed resource "
        "that are not PolicySetComponents and, therefore, have no "
        "other relative priority defined. The priority is a "
        "non-negative integer; a larger value indicates a higher "
        "priority.") ]
    uint16 Priority;
};

// ==================================================================
// PolicySetValidityPeriod
// ==================================================================
    [Association, Aggregation, Version ("2.7.0"), Description (
        "The PolicySetValidityPeriod aggregation represents scheduled "
        "activation and deactivation of a PolicySet. A PolicySet is "
        "considered \"active\" if it is both \"Enabled\" and in a "
        "valid time period. \n"
        "If a PolicySet is associated with multiple policy time "
        "periods via this association, then the Set is in a valid "
        "time period if at least one of the time periods evaluates to "
        "TRUE. If a PolicySet is contained in another PolicySet via "
        "the PolicySetComponent aggregation (e.g., a PolicyRule in a "
        "PolicyGroup), then the contained PolicySet (e.g., "
        "PolicyRule) is in a valid period if at least one of the "
        "aggregate’s PolicyTimePeriodCondition instances evaluates to "
        "TRUE and at least one of its own PolicyTimePeriodCondition "
        "instances also evalutes to TRUE. (In other words, the "
        "PolicyTimePeriodConditions are OR’ed to determine whether "
        "the PolicySet is in a valid time period and then AND’ed with "
        "the OR’ed PolicyTimePeriodConditions of each of PolicySet "
        "instances in the PolicySetComponent hierarchy to determine "
        "if the PolicySet is in a valid time period and, if also "
        "\"Enabled\", therefore, active; i.e., the hierachy ANDs the "
        "OR’ed PolicyTimePeriodConditions of the elements of the "
        "hierarchy. \n"
        "A Time Period may be aggregated by multiple PolicySets. A "
        "Set that does not point to a PolicyTimePeriodCondition via "
        "this association, from the point of view of scheduling, is "
        "always in a valid time period.") ]
class CIM_PolicySetValidityPeriod : CIM_PolicyComponent {

    [Aggregate, Override ( "GroupComponent"), Description (
        "This property contains the name of a PolicySet that "
        "contains one or more PolicyTimePeriodConditions.") ]
    CIM_PolicySet REF GroupComponent;

    [Override ( "PartComponent"), Description (
        "This property contains the name of a "
        "PolicyTimePeriodCondition defining the valid time periods "
        "for one or more PolicySets.") ]
    CIM_PolicyTimePeriodCondition REF PartComponent;
};

// ==================================================================
// PolicyConditionStructure
// ==================================================================
    [Association, Abstract, Aggregation, Version ("2.7.0"),
    Description (
        "PolicyConditions may be aggregated into rules and into "
        "compound conditions. PolicyConditionStructure is the "
        "abstract aggregation class for the structuring of policy "
        "conditions. \n"
        "The Conditions aggregated by a PolicyRule or "
        "CompoundPolicyCondition are grouped into two levels of "
        "lists: either an OR’ed set of AND’ed sets of conditions "
        "(DNF, the default) or an AND’ed set of OR’ed sets of "
        "conditions (CNF). Individual PolicyConditions in these lists "
        "may be negated. The property ConditionListType specifies "
        "which of these two grouping schemes applies to a particular "
        "PolicyRule or CompoundPolicyCondition instance. \n"
        "One or more PolicyTimePeriodConditions may be among the "
        "conditions associated with a PolicyRule or "
        "CompoundPolicyCondition via the PolicyConditionStructure "
        "subclass association. In this case, the time periods are "
        "simply additional Conditions to be evaluated along with any "
        "others that are specified.") ]
class CIM_PolicyConditionStructure : CIM_PolicyComponent {

    [Aggregate, Override ( "GroupComponent"), Description (
        "This property represents the Policy that contains one or "
        "more PolicyConditions.") ]
    CIM_Policy REF GroupComponent;

    [Override ( "PartComponent"), Description (
        "This property holds the name of a PolicyCondition contained "
        "by one or more PolicyRule or CompoundPolicyCondition "
        "instances.") ]
    CIM_PolicyCondition REF PartComponent;

    [Description (
        "Unsigned integer indicating the group to which the "
        "contained PolicyCondition belongs. This integer segments "
        "the Conditions into the AND’ed sets (when the "
        "ConditionListType is \"DNF\") or, similarly, into the OR’ed "
        "sets (when the ConditionListType is \"CNF\").") ]
    uint16 GroupNumber;

    [Description (
        "Indication of whether the contained PolicyCondition is "
        "negated. TRUE indicates that the PolicyCondition IS "
        "negated, FALSE indicates that it IS NOT negated.") ]
    boolean ConditionNegated;
};

// ==================================================================
// PolicyActionStructure
// ==================================================================
    [Association, Abstract, Aggregation, Version ("2.6.0"),
    Description (
        "PolicyActions may be aggregated into rules and into compound "
        "actions. PolicyActionStructure is the abstract aggregation "
        "class for the structuring of policy actions.") ]
class CIM_PolicyActionStructure : CIM_PolicyComponent {

    [Aggregate, Override ( "GroupComponent"), Description (
        "PolicyAction instances may be aggregated into either "
        "PolicyRule instances or CompoundPolicyAction instances.") ]
    CIM_Policy REF GroupComponent;

    [Override ( "PartComponent"), Description (
        "A PolicyAction aggregated by a PolicyRule or "
        "CompoundPolicyAction.") ]
    CIM_PolicyAction REF PartComponent;
    [Description (
        "ActionOrder is an unsigned integer 'n' that indicates the "
        "relative position of a PolicyAction in the sequence of "
        "actions associated with a PolicyRule or "
        "CompoundPolicyAction. When 'n' is a positive integer, it "
        "indicates a place in the sequence of actions to be "
        "performed, with smaller integers indicating earlier "
        "positions in the sequence. The special value '0' indicates "
        "'don't care'. If two or more PolicyActions have the same "
        "non-zero sequence number, they may be performed in any "
        "order, but they must all be performed at the appropriate "
        "place in the overall action sequence. \n"
        "A series of examples will make ordering of PolicyActions "
        "clearer: \n"
        "- If all actions have the same sequence number, regardless "
        "  of whether it is '0' or non-zero, any order is "
        "  acceptable. \n"
        "- The values: \n"
        "    1:ACTION A \n"
        "    2:ACTION B \n"
        "    1:ACTION C \n"
        "    3:ACTION D \n"
        "  indicate two acceptable orders: A,C,B,D or C,A,B,D, "
        "  since A and C can be performed in either order, but only "
        "  at the '1' position. \n"
        "- The values: \n"
        "    0:ACTION A \n"
        "    2:ACTION B \n"
        "    3:ACTION C \n"
        "    3:ACTION D \n"
        "  require that B,C, and D occur either as B,C,D or as B,D,C. "
        "  Action A may appear at any point relative to B, C, and "
        "  D. \n"
        "Thus the complete set of acceptable orders is: A,B,C,D; "
        "B,A,C,D; B,C,A,D; B,C,D,A; A,B,D,C; B,A,D,C; B,D,A,C; "
        "B,D,C,A. \n"
        "Note that the non-zero sequence numbers need not start with "
        "'1', and they need not be consecutive. All that matters is "
        "their relative magnitude.") ]
    uint16 ActionOrder;
};

// ==================================================================
// PolicySetAppliesToElement
// ==================================================================
    [Association, Experimental, Version ("2.7.1000"), Description (
        "PolicySetAppliesToElement makes explicit which PolicySets "
        "(i.e., policy rules and groups of rules) ARE CURRENTLY "
        "applied to a particular Element. This association indicates "
        "that the PolicySets that are appropriate for a "
        "ManagedElement (specified using the PolicyRoleCollection "
        "aggregation) have actually been deployed in the policy "
        "management infrastructure. Note that if the named Element "
        "refers to a Collection, then the PolicySet is assumed to be "
        "applied to all the members of the Collection.") ]
class CIM_PolicySetAppliesToElement {

    [Key, Description (
        "The PolicyRules and/or groups of rules that are currently "
        "applied to an Element.") ]
    CIM_PolicySet REF PolicySet;

    [Key, Description (
        "The ManagedElement to which the PolicySet applies.") ]
    CIM_ManagedElement REF ManagedElement;
};

// ==================================================================
// end of file
// ==================================================================
```