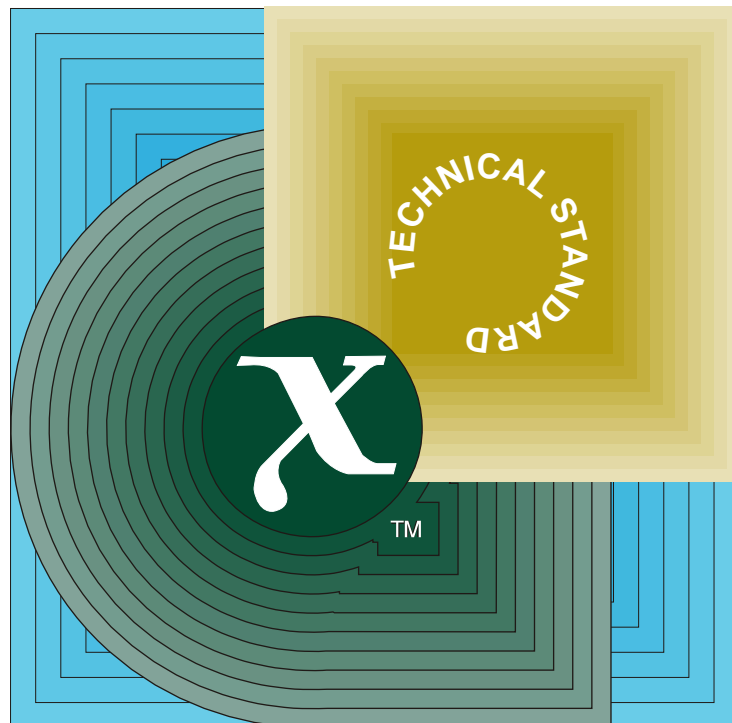


Technical Standard

Systems Management: GDMO to XOM Translation Algorithm



THE *Open* GROUP

[This page intentionally left blank]

X/Open CAE Specification

Systems Management: GDMO to XOM Translation Algorithm

X/Open Company Ltd.



© October 1995, X/Open Company Limited

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

X/Open CAE Specification

Systems Management: GDMO to XOM Translation Algorithm

ISBN: ISBN 1-85912-126-8

X/Open Document Number: C502

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to X/Open at:

X/Open Company Limited
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

XoSpecs@xopen.org

Contents

Chapter	1	Introduction.....	1
	1.1	Background.....	1
	1.2	Purpose	2
	1.3	Scope.....	3
	1.4	Future Directions	4
Chapter	2	Algorithm Input Specification	5
	2.1	OM Package Template	5
	2.2	ISO GDMO Templates	6
	2.3	ASN.1 Syntax Modules.....	9
	2.4	Input Assumptions.....	10
	2.4.1	OSI Object Identifiers Must Include Decimal Values.....	10
	2.4.2	IMPORTed Data Types Must be Included with Input	10
	2.4.3	Uniqueness of OM Package Template Input.....	10
	2.5	Possible Input Extensions	11
	2.5.1	Alternative Input Formats	11
	2.5.2	Practical Implementation Restrictions	11
Chapter	3	Algorithm Output Specification	13
	3.1	OM Package Definitions.....	15
	3.1.1	OM Package Object Identifier	15
	3.1.2	Object Identifier Tables	15
	3.1.3	Information Syntax Tables	16
	3.1.4	OM Attribute Tables.....	17
	3.1.5	Value Lists	18
	3.2	C Header File	19
	3.2.1	OM Package Object Identifier	19
	3.2.2	Object Identifier Definitions	19
	3.2.3	OM Class Constants	20
	3.3	Encoding Definitions.....	21
	3.3.1	OM Class Encoding Definitions.....	21
	3.3.2	OM Attribute Encoding Definitions	21
	3.4	Workspace Contents Package Output Format	23
	3.5	Output Constraints.....	24
	3.5.1	Reuse of Existing Package Definitions	24
	3.6	Possible Output Extensions	25
	3.6.1	No Abstract OM Classes Generated.....	25
	3.6.2	Updating Existing OM Packages.....	25
	3.6.3	Single Intermediate Output Format.....	25

Chapter 4	Detailed Translation Algorithm	27
4.1	OM Package Template	28
4.1.1	Input Structure	28
4.1.2	Output Structures	29
4.1.3	Translation Rules	29
4.1.4	Example	30
4.2	Managed Object Class Template	31
4.2.1	Input Structure	31
4.2.2	Output Structures	31
4.2.3	Translation Rules	31
4.2.4	Example	32
4.3	Parameter Template	33
4.3.1	Input Structure	33
4.3.2	Output Structures	33
4.3.3	Translation Rules	33
4.3.4	Example	34
4.4	Attribute Template	36
4.4.1	Input Structure	36
4.4.2	Output Structures	36
4.4.3	Translation Rules	36
4.4.4	Example	37
4.5	Attribute Group Template	39
4.5.1	Input Structure	39
4.5.2	Output Structures	39
4.5.3	Translation Rules	39
4.5.4	Example	39
4.6	Action Template	41
4.6.1	Input Structure	41
4.6.2	Output Structures	41
4.6.3	Translation Rules	41
4.6.4	Example	42
4.7	Notification Template	43
4.7.1	Input Structure	43
4.7.2	Output Structures	43
4.7.3	Translation Rules	43
4.7.4	Example	44
4.8	ASN.1 Module	46
4.8.1	Input Structure	46
4.8.2	Output Structures	46
4.8.3	Translation Rules	47
4.8.4	Example	54
4.9	Workspace Contents Package Output Translation.....	57
4.9.1	Input Structure	57
4.9.2	Output Structure	57
4.9.3	Translation Rules	58
4.9.4	Example	59
4.10	Summary of the Translation Rules.....	60
4.10.1	Rules for OM Package Definition and C Header Files.....	61

- 4.10.2 Rules for C Header Files Only..... 62
- 4.10.3 Rules for Encoding Definitions Only..... 62

- Appendix A**
- DMI Contents Package 63**
- A.1 Attributes and Event Types 65
- A.2 Object Identifiers for DMI..... 66
 - A.2.1 DMI OM Package Object Identifier..... 66
 - A.2.2 Object Identifiers for DMI Object Classes 66
 - A.2.3 Object Identifiers for DMI Attributes 66
 - A.2.4 Object Identifiers for DMI Attribute Groups 68
 - A.2.5 Object Identifiers for DMI Notifications 68
 - A.2.6 Object Identifiers for DMI Parameters 69
- A.3 DMI Attribute Value Syntaxes 70
- A.4 DMI Notification Information Syntaxes 73
- A.5 DMI Parameter Syntaxes..... 74
- A.6 OM Class Hierarchy and Attribute Tables 75
- A.7 Attribute Enumeration Lists..... 94
- A.8 C Header File 100
- A.9 Workspace Content Package Output 109
- A.10 Encoding Definitions..... 117

- Glossary 127**

- Index..... 131**

List of Figures

1-1	GDMO to OM Algorithm Architecture.....	3
3-1	GDMO to OM Algorithm Architecture Details.....	13
4-1	Overall Translation Process Flow.....	48
4-2	GDMO to OM Algorithm Inputs and Outputs	60

List of Tables

3-1	Object Identifier Usage	15
3-2	OM Attributes with Syntax “ANY”	16
4-1	Valid ASN.1 Syntax Type Definitions.....	49
4-2	Valid OM Class and OM Attribute Name Constants.....	49
A-1	OM Attributes of Additional-Information	77
A-2	OM Attributes of Alarm-Info	77
A-3	OM Attributes of Alarm-Status	77
A-4	OM Attributes of Allomorphs.....	78
A-5	OM Attributes of Attribute-Identifier-List	78
A-6	OM Attributes of Attribute-List.....	78
A-7	OM Attributes of Attribute-Value-Change-Definition	78
A-8	OM Attributes of Attribute-Change-Info	79
A-9	OM Attributes of Availability-Status	79
A-10	OM Attributes of Back-Up-Destination-List.....	79
A-11	OM Attributes of Back-Up-Relationship-Object	79
A-12	OM Attributes of Capacity-Alarm-Threshold.....	80
A-13	OM Attributes of Control-Status.....	80
A-14	OM Attributes of Correlated-Notifications.....	80
A-15	OM Attributes of Correlated-Notifications-1	80
A-16	OM Attributes of Counter-Threshold	81
A-17	OM Attributes of Destination	81
A-18	OM Attributes of Down	81
A-19	OM Attributes of Gauge-Threshold.....	81
A-20	OM Attributes of Group-Objects.....	82
A-21	OM Attributes of Intervals-of-Day	82
A-22	OM Attributes of Management-Extension	82
A-23	OM Attributes of Monitored-Attributes.....	82
A-24	OM Attributes of AE-Title	83
A-25	OM Attributes of Notify-Threshold.....	83
A-26	OM Attributes of Object-Info	83
A-27	OM Attributes of Observed-Value	83
A-28	OM Attributes of Packages.....	84
A-29	OM Attributes of Prioritised-Object	84
A-30	OM Attributes of Probable-Cause.....	84
A-31	OM Attributes of Procedural-Status.....	85
A-32	OM Attributes of Proposed-Repair-Actions	85
A-33	OM Attributes of Relationship-Change-Info	85
A-34	OM Attributes of Security-Alarm-Detector	85
A-35	OM Attributes of Security-Alarm-Info.....	86

Contents

A-36 OM Attributes of Service-User..... 86
A-37 OM Attributes of Setof-Attribute-Value-Change-Definition..... 86
A-38 OM Attributes of Setof-Correlated-Notifications 86
A-39 OM Attributes of Setof-Counter-Threshold..... 87
A-40 OM Attributes of Setof-Gauge-Threshold..... 87
A-41 OM Attributes of Setof-Intervals-Of-Day 87
A-42 OM Attributes of Setof-Prioritised-Object..... 87
A-43 OM Attributes of Setof-Supported-Features..... 88
A-44 OM Attributes of Setof-Week-Mask 88
A-45 OM Attributes of Simple-Name-Type..... 88
A-46 OM Attributes of Specific-Identifier 88
A-47 OM Attributes of Specific-Problems..... 89
A-48 OM Attributes of State-Change-Info 89
A-49 OM Attributes of Stop-Time..... 89
A-50 OM Attributes of Supported-Feature 90
A-51 OM Attributes of System-Id 91
A-52 OM Attributes of System-Title..... 91
A-53 OM Attributes of Threshold-Info 91
A-54 OM Attributes of Threshold-Level-Ind..... 92
A-55 OM Attributes of Tide-Mark 92
A-56 OM Attributes of Tide-Mark-Info 92
A-57 OM Attributes of Time24 92
A-58 OM Attributes of Up 93
A-59 OM Attributes of Week-Mask..... 93

Preface

X/Open

X/Open is an independent, worldwide, open systems organisation supported by most of the world's largest information systems suppliers, user organisations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems.

X/Open's strategy for achieving this goal is to combine existing and emerging standards into a comprehensive, integrated, high-value and usable open system environment, called the Common Applications Environment (CAE). This environment covers the standards, above the hardware level, that are needed to support open systems. It provides for portability and interoperability of applications, and so protects investment in existing software while enabling additions and enhancements. It also allows users to move between systems with a minimum of retraining.

X/Open defines this CAE in a set of specifications which include an evolving portfolio of application programming interfaces (APIs) which significantly enhance portability of application programs at the source code level, along with definitions of and references to protocols and protocol profiles which significantly enhance the interoperability of applications and systems.

The X/Open CAE is implemented in real products and recognised by a distinctive trade mark — the X/Open brand — that is licensed by X/Open and may be used on products which have demonstrated their conformance.

X/Open Technical Publications

X/Open publishes a wide range of technical literature, the main part of which is focussed on specification development, but which also includes Guides, Snapshots, Technical Studies, Branding/Testing documents, industry surveys, and business titles.

There are two types of X/Open specification:

- *CAE Specifications*

CAE (Common Applications Environment) specifications are the stable specifications that form the basis for X/Open-branded products. These specifications are intended to be used widely within the industry for product development and procurement purposes.

Anyone developing products that implement an X/Open CAE specification can enjoy the benefits of a single, widely supported standard. In addition, they can demonstrate compliance with the majority of X/Open CAE specifications once these specifications are referenced in an X/Open component or profile definition and included in the X/Open branding programme.

CAE specifications are published as soon as they are developed, not published to coincide with the launch of a particular X/Open brand. By making its specifications available in this way, X/Open makes it possible for conformant products to be developed as soon as is practicable, so enhancing the value of the X/Open brand as a procurement aid to users.

- *Preliminary Specifications*

These specifications, which often address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations, are released in a controlled manner for the purpose of validation through implementation of products. A Preliminary specification is not a draft specification. In fact, it is as stable as X/Open can make it, and on publication has gone through the same rigorous X/Open development and review procedures as a CAE specification.

Preliminary specifications are analogous to the *trial-use* standards issued by formal standards organisations, and product development teams are encouraged to develop products on the basis of them. However, because of the nature of the technology that a Preliminary specification is addressing, it may be untried in multiple independent implementations, and may therefore change before being published as a CAE specification. There is always the intent to progress to a corresponding CAE specification, but the ability to do so depends on consensus among X/Open members. In all cases, any resulting CAE specification is made as upwards-compatible as possible. However, complete upwards-compatibility from the Preliminary to the CAE specification cannot be guaranteed.

In addition, X/Open publishes:

- *Guides*

These provide information that X/Open believes is useful in the evaluation, procurement, development or management of open systems, particularly those that are X/Open-compliant. X/Open Guides are advisory, not normative, and should not be referenced for purposes of specifying or claiming X/Open conformance.

- *Technical Studies*

X/Open Technical Studies present results of analyses performed by X/Open on subjects of interest in areas relevant to X/Open's Technical Programme. They are intended to communicate the findings to the outside world and, where appropriate, stimulate discussion and actions by other bodies and the industry in general.

- *Snapshots*

These provide a mechanism for X/Open to disseminate information on its current direction and thinking, in advance of possible development of a Specification, Guide or Technical Study. The intention is to stimulate industry debate and prototyping, and solicit feedback. A Snapshot represents the interim results of an X/Open technical activity. Although at the time of its publication, there may be an intention to progress the activity towards publication of a Specification, Guide or Technical Study, X/Open is a consensus organisation, and makes no commitment regarding future development and further publication. Similarly, a Snapshot does not represent any commitment by X/Open members to develop any specific products.

Versions and Issues of Specifications

As with all *live* documents, CAE Specifications require revision, in this case as the subject technology develops and to align with emerging associated international standards. X/Open makes a distinction between revised specifications which are fully backward compatible and those which are not:

- a new *Version* indicates that this publication includes all the same (unchanged) definitive information from the previous publication of that title, but also includes extensions or additional information. As such, it *replaces* the previous publication.

- a new *Issue* does include changes to the definitive information contained in the previous publication of that title (and may also include extensions or additional information). As such, X/Open maintains *both* the previous and new issue as current publications.

Corrigenda

Most X/Open publications deal with technology at the leading edge of open systems development. Feedback from implementation experience gained from using these publications occasionally uncovers errors or inconsistencies. Significant errors or recommended solutions to reported problems are communicated by means of Corrigenda.

The reader of this document is advised to check periodically if any Corrigenda apply to this publication. This may be done either by email to the X/Open info-server or by checking the Corrigenda list in the latest X/Open Publications Price List.

To request Corrigenda information by email, send a message to `info-server@xopen.co.uk` with the following in the Subject line:

```
request corrigenda; topic index
```

This will return the index of publications for which Corrigenda exist.

This Document

X/Open defines an industry standard Application Programming Interface (API) to support management communication in two integrally related specifications:

- reference **XOM**
- reference **XMP**.

These specifications collectively define a single API to support management communication offered by the Open Systems Interconnection (OSI) Common Management Information Service (CMIS) and the Internet Simple Network Management Protocol (SNMP). However, these specifications currently do not provide for automated creation of management content-specific packages which applications require to make use of the API.

This document specifies an algorithm for deriving content-specific API extensions in OM format from information model specifications in the International Organization for Standardization (ISO) Guidelines for the Definition of Managed Objects template format. This algorithm presents a solution to issues facing organisations which plan to define and/or implement information models using the XMP API. The translation algorithm:

- Provides a straight-forward mechanism to develop OM packages to support numerous information management information models.
- Defines a standard, vendor-independent mapping between an information model and the supporting API, permitting content-specific data to be handled in the API without sacrificing vendor independence.
- Relieves an application developer from using a vendor ASN.1 compiler, by enabling the API to perform encoding and decoding of data which is information-model-specific.
- Provides consistency across the programming interface. An application builds content-independent and content-dependent management service parameters using the same set of XOM routines.
- Allows automation, through development of tools which accept management information models as input and produce content-specific API extensions as output.

Note: OM modules derived from different levels of object definitions in GDMO are not necessarily interoperable. The identical level of the object definition should be verified to ensure interoperability. For example, ISO/IEC 10165-2 (DMI) is likely to be changed by an ISO Technical Corrigendum. This will cause the original DMI OM modules not to be fully interoperable with the new DMI OM modules reflecting the ISO Technical Corrigendum changes.

Structure

- Chapter 1 explains the positioning, scope, purpose and status of this specification.
- Chapter 2 describes the inputs which are required to drive the translation algorithm. It includes examples from a sample DMI information model. Readers already familiar with GDMO can skip to Chapter 4.
- Chapter 3 describes the OM package output generated by the translation algorithm. It also covers output constraints that are placed on the algorithm, and possible extensions which are beyond the current scope of this algorithm. Again, it includes examples from a sample DMI information model. Readers already familiar with GDMO can skip to Chapter 4.
- Chapter 4 specifies the translation algorithm for generating X/Open OM packages from ISO GDMO template inputs, detailing the translation rules and input and output structures involved. It includes examples from a sample DMI information model.
- Appendix A contains the full translation of the DMI Contents Package (reference **DMI**) for use with X/Open's System Management: Management Protocol API (reference **XMP**). This DMI package has been generated by applying the GDMO-XOM translation algorithm to the GDMO definitions referenced in this Appendix. It is provided for illustrative purposes. In the case of any conflict between the definition of the algorithm, and the example output provided in this appendix, the algorithm is to be taken as authoritative.

The document concludes with a glossary and an index.

The examples used in Chapters 2 through 4 are all drawn from a sample information model specified by ISO/IEC 10165-2, Definition of Management Information (reference **DMI**), which is presented in Appendix A.

Typographical Conventions

The following typographical conventions are used in this document, except where defined otherwise:

- **Bold** font is used in text for options to commands, filenames, keywords, type names, data structures and their members, and language-independent names.
- *Italic* strings are used for emphasis or to identify the first instance of a word requiring definition. Italics in text also denote:
 - command operands, command option-arguments or variable names, for example, substitutable argument prototypes
 - environment variables, which are also shown in capitals
 - utility names
 - external variables, such as *errno*
 - functions; these are shown as follows: *name()*. Names without parentheses are C external variables, C function family names, utility names, command operands or command

option-arguments.

- Roman font is used for the names of constants and literals.
- The notation `<file.h>` indicates a header file.
- Names surrounded by braces, for example, `{ARG_MAX}`, represent symbolic limits or configuration values which may be declared in appropriate headers by means of the C `#define` construct.
- The notation `[EABCD]` is used to identify a return value `ABCD`, including if this is an error value.
- Syntax, code examples and user input in interactive examples are shown in fixed width font. Brackets shown in this font, `[]`, are part of the syntax and do *not* indicate optional items.

Trade Marks

X/Open[®] is a registered trade mark, and the “X” device is a trade mark, of X/Open Company Limited.

Referenced Documents

The following documents are referenced in this specification:

ASN.1

ISO 8824: 1990 Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).

BER

ISO/IEC 8825: 1990 (ITU-T Recommendation X.209 (1988)), Information Technology — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

DMI

ISO/IEC 10165-2: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 2: Definition of Management Information.

GDMO

ISO/IEC 10165-4: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 4: Guidelines for the Definition of Managed Objects. of Managed Objects, July 1991.

NMF Strategy for Migration to GDMO

OSI/NM Forum Strategy for Migration to GDMO; Forum 010, Issue 1.0, January 1991.

SMF

ISO/IEC 10164: 1992 Information Technology — Open Systems Interconnection — Systems Management (Parts 1 to 13 inclusive).

XMP

X/Open CAE Specification, March 1994, Systems Management: Management Protocol API (ISBN 1-85912-027-X, C306).

XOM, Issue 2

X/Open CAE Specification, February 1994, OSI-Abstract-Data Manipulation API (XOM), Issue 2 (ISBN: 1-85912-008-3, C315).

1.1 Background

X/Open defines an industry standard Application Programming Interface (API) to support management communication in two integrally related specifications:

- reference **XOM**
- reference **XMP**.

These specifications collectively define a single API to support management communication offered by the Open Systems Interconnection (OSI) Common Management Information Service (CMIS) and the Internet Simple Network Management Protocol (SNMP). However, these specifications currently do not provide for automated creation of management content-specific packages which applications require to make use of the API.

This specification addresses the creation of management content-specific packages via a mechanised algorithm. The perspective and direction that this algorithm focuses on is conformance to consistency with the XOM API and the XMP API style of specifying Object Management (OM) objects. Any exceptions or deviations from this style are noted throughout this document. There are many ways to generate OM packages from the Guidelines for the Definition of Managed Objects (GDMO) templates (reference **GDMO**), and there are many variations to the algorithm specified in this document. The approach that this algorithm takes is one that is consistent with the style that is defined in the XOM API and the XMP API.

1.2 Purpose

This specification specifies an algorithm for deriving content-specific API extensions in OM format from information model specifications in the International Organization for Standardization (ISO) Guidelines for the Definition of Managed Objects template format. This algorithm presents a solution to problems facing organisations which plan to define and/or implement information models using the XMP API. These problems include:

- An additional OM package is required to support each new information model. The OSI/Network Management Forum (NMF) package definition (reference **NMF Migration to GDMO**) is an example of such an extension. Another package has recently been developed to support the ISO Definition of Management Information (reference **DMI**). Unless a standard translation algorithm is published, every standard information model will require publication of a standard package definition if it is to be supported by the XMP API.
- The XMP mechanism for supplying content-specific data (as Abstract Syntax Notation One (ASN.1) Basic Encoding Rules (BER) encoded values) is more complicated than and inconsistent with the XMP mechanism for supplying content-independent data (as OM packages). It is therefore advantageous to represent new information models as new content-specific OM packages to provide a consistent presentation of attribute values, event information and results, action information and results, and parameter information to the application. The algorithm defined in this document standardises and therefore helps to automate the production of the content-specific OM packages.

The translation algorithm defined in this document solves these problems by producing a consistent, extensible, vendor-independent OM package for any information model. The algorithm defines a deterministic mapping between the information model in ISO GDMO templates and the X/Open OM package definitions necessary to support that information model. Specifically, it:

- provides a straight-forward mechanism to develop OM packages to support numerous management information models
- defines a standard, vendor-independent mapping between an information model and the supporting API, permitting content-specific data to be handled in the API without sacrificing vendor independence
- relieves an application developer from using a vendor ASN.1 compiler, by enabling the API to perform encoding and decoding of data which is information-model-specific
- provides consistency across the programming interface. An application builds content-independent and content-dependent management service parameters using the same set of XOM routines
- allows automation, through development of tools which accept management information models as input and produce content-specific API extensions as output.

1.3 Scope

This specification first summarises the input which drives the algorithm - ISO GDMO templates, Chapter 2 - and the output generated by the algorithm - X/Open OM packages, cX chap3 . 1 It then specifies the translation algorithm to convert the ISO GDMO template inputs to the X/Open OM package outputs. The translation rules, defined in Chapter 4, describe how the information model input is converted to the OM package outputs.

Figure 1-1 illustrates the high-level architecture of the algorithm in terms of its inputs and outputs. The input is the Information Model Specification, in ISO GDMO template format. The output is the OM package.

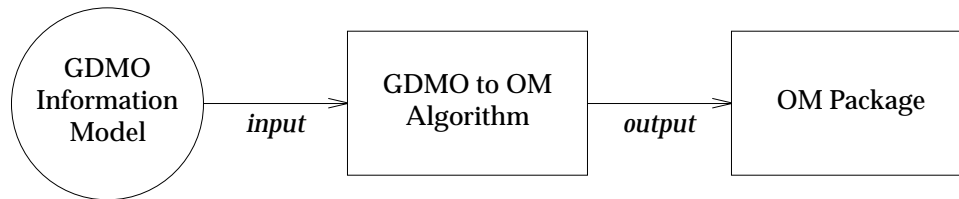


Figure 1-1 GDMO to OM Algorithm Architecture

The input, ISO GDMO templates described in reference **GDMO**, provides developers of managed object class definitions with the information and the documentation tools required to produce complete managed object class definitions that are compatible with the OSI management standards developed jointly by ISO/IEC and CCITT. The output, X/Open OM packages described in the XMP API, consists of content-specific OM classes representing notification types, attribute types, action types, and parameter types which are exchanged between the management processes (that is, the contents of notifications and operations).

This document assumes the reader has read and is familiar with the reference **XOM**, the reference **XMP**, and reference **GDMO**.

The structure of this document is described in the Preface.

Note: Appendix A contains the full translation of the DMI Contents Package (reference **DMI** for use with X/Open's System Management: Management Protocol API (reference **XMP**). This DMI package is provided for illustrative purposes.

In the case of any conflict between the definition of the algorithm, and the example output provided in Appendix A, the algorithm is to be taken as authoritative.

1.4 Future Directions

Areas where there are likely to be future developments include:

- translation of certain complex ASN.1 type definitions (for example, SEQUENCE OF SET OF SEQUENCE)
- generation of unambiguous OM class and attribute labels.

These issues are further discussed in Section 4.8.3.

Algorithm Input Specification

This chapter provides a summary description of the inputs which are required to drive the translation algorithm. There are three types of input required:

- an OM package template
- a set of ISO GDMO templates
- all referenced ASN.1 syntax module(s).

This chapter concludes with a description of input assumptions that are placed on the algorithm, and with a brief discussion of possible input extensions which were also considered during the development of this algorithm but in terms of this specification are currently beyond the scope of the algorithm.

Throughout this chapter, examples are provided from a sample DMI information model; see reference **DMI** for the complete sample input specification.

2.1 OM Package Template

An OM package is a set of OM classes and OM attributes that are grouped together by specification (for example, OM classes and OM attributes which represent an information model). In addition to information already available in GDMO templates and ASN.1 modules, there are a few pieces of additional information that the algorithm requires to generate unique OM packages. This information can be provided as input in the following template OBformat; refer to Section 4.1 on page 28 for additional discussion.

```
SMI_PART2 OM-PACKAGE
  ABBREVIATION DMI
  INITIAL ATTRIBUTE VALUE 11301
  IMPORTED OM-PACKAGES
    {iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) common(1)},
    {iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) cmis(2)}
  REFERENCE "This package translates all templates defined by Rec. X.721 | ISO/IEC 10165-2 : 1992"
  ::= {iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) dmi(4)}
```

2.2 ISO GDMO Templates

The algorithm specified in this document requires input in the form of ISO GDMO templates defined by reference **GDMO**. Input templates are summarised in the following headings, each illustrated by an example drawn from the sample DMI information model in reference **DMI**.

Managed Object Class Template

This forms the basis of formal definition of an ISO GDMO managed object, for example:

```

system MANAGED OBJECT CLASS
  DERIVED FROM top
  CHARACTERIZED BY
  systemPackage PACKAGE
  ATTRIBUTES
    systemId      GET,
    systemTitle   GET,
    operationalState  GET,
    usageState    GET;;;
  CONDITIONAL PACKAGES
  administrativeStatePackage PACKAGE
  ATTRIBUTES
    administrativeState  GET-REPLACE;
  REGISTERED AS { smi2Package 14 };
    PRESENT IF "an instance supports it.",
  supportedFeaturesPackage PACKAGE
  ATTRIBUTES
    supportedFeatures  GET-REPLACE ADD-REMOVE;
  REGISTERED AS { smi2Package 15 };
    PRESENT IF "an instance supports it.";
  REGISTERED AS { smi2Package 13 };

```

Package Template

This contains a collection of behaviour, attributes, attribute groups, operations, notifications and parameters which can be inserted into managed object class templates using the CHARACTERIZED BY or CONDITIONAL PACKAGES, for example:

```

duration PACKAGE
  ATTRIBUTES
    startTime  GET-REPLACE,
    stopTime  REPLACE-WITH-DEFAULT
              DEFAULT VALUE
  Attribute-ASN1Module.defaultStop Time
    GET-REPLACE;
  REGISTERED AS { smi2MObjectClass 26 };

```

Parameter Template

This is used to define syntaxes and behaviours associated with particular attributes, operations and notifications, expanding ANY DEFINED BY syntaxes which appear in other definitions, for example:


```

miscellaneousError PARAMETER
  CONTEXT SPECIFIC-ERROR
  WITH SYNTAX Parameter-ASN1Module.MiscellaneousError;
  BEHAVIOUR miscellaneousErrorBehaviour
  BEHAVIOUR DEFINED AS "When a processing error failure has occurred
    and the error condition does not match any of the object's
    defined specific error types, this value is used.";
  REGISTERED AS { smi2Parameter 1 };

```

Name Binding Template

This allows alternative naming structures to be defined for managed objects, for example:

```

discriminator-system NAME BINDING
  SUBORDINATE OBJECT CLASS discriminator AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS system AND SUBCLASSES;
  WITH ATTRIBUTE discriminatorId;
  CREATE WITH-REFERENCE-OBJECT,
    WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
  REGISTERED AS { smi2NameBinding 1 }

```

Attribute Template

This is used to define individual attribute types, for example:

```

discriminatorId ATTRIBUTE
  WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.SimpleName Type;
  MATCHES FOR EQUALITY, SUBSTRINGS, ORDERING;
  BEHAVIOUR rDNIdBehaviour;
  REGISTERED AS { smi2AttributeID 1 };

```

Attribute Group Template

This allows attribute groupings to be defined as short-hand references which may be included in get attribute or set-to-default requests, for example:

```

state ATTRIBUTE GROUP
  DESCRIPTION "This is defined as an empty group. The elements of
    this group are composed of state attributes in the managed object.
    The state attributes may include those specified in CCITT Rec. X.731
    | ISO/IEC 10164-2 and others that are specific to the managed object class.";
  REGISTERED AS { smi2AttributeGroup 1 };

```

Behaviour Template

This is used to define behavioural aspects and convey semantics that are not completely defined by other templates, for example:

```

attributeValueChangeDefinitionBehaviour BEHAVIOUR
  DEFINED AS "This attribute contains a set of attribute
    identifiers and their old and new values";

```

Action Template

This is used to define the behaviour and syntax associated with a particular Action type. There are no ACTION templates defined by the sample input information model ISO/IEC 10165-2.

Notification Template

This is used to define the behaviour and syntax associated with a particular Notification type, for example:

```

attributeValueChange NOTIFICATION
  BEHAVIOUR attributeValueChangeBehaviour;
  WITH INFORMATION SYNTAX
  Notification-ASN1Module.AttributeValueChangeInfo
  AND ATTRIBUTE IDS
  sourceIndicator
  attributeIdentifierList
  attributeValueChangeDefinition
  notificationIdentifier
  correlatedNotifications
  additionalText
  additionalInformation
REGISTERED AS { smi2Notification 1 };
sourceIndicator,
attributeIdentifierList,
attributeValueChangeDefinition,
notificationIdentifier,
correlatedNotifications,
additionalText,
additionalInformation;
```

2.3 ASN.1 Syntax Modules

ASN.1 type or value definitions referenced by GDMO templates representing an information module are required as input to the algorithm defined in this document. In many cases, the name of the ASN.1 type or value is prefixed by an ASN.1 syntax module name which contains the type or value definition. When no such prefix is used, the name is assumed to refer to an ASN.1 module in the same document that contains the template. For example:

```
Parameter-ASN1Module { joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 3 }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
    smi2Parameter OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) smi(3) part2(2) parameter(5) }
    MiscellaneousError ::= NULL
END
```

This sample module defines an object identifier referenced wherever the ASN.1 label *smi2Parameter* appears in a template. In the example given previously for the PARAMETER template:

```
“REGISTERED AS { smi2Parameter 1 };”
```

equates to:

```
“REGISTERED AS { joint-iso-ccitt ms(9) smi(3) part2(2) parameter(5) 1 };”.
```

This sample module also defines the type referenced by the same PARAMETER template’s “WITH SYNTAX Parameter-ASN1Module.MiscellaneousError;” clause.

Refer to reference **ASN.1** for additional information regarding the format of ASN.1 syntax modules. An information model must include one or more ASN.1 modules which fully define all referenced ASN.1 type definitions, including those referenced in GDMO ATTRIBUTE, NOTIFICATION, ACTION or PARAMETER template “SYNTAX” clauses, and object identifiers referenced in GDMO template “REGISTERED AS” clauses. Occasionally, referenced ASN.1 type definitions will be IMPORTed from another ASN.1 module. The algorithm defined in this document requires as input all referenced ASN.1 type definitions, including those IMPORTed from other modules from within the same package.

2.4 Input Assumptions

The following assumptions are imposed on input specifications accepted by the algorithm defined in this document. The assumptions imply the amount of information required as input for the algorithm to perform its job, and are not intended to imply changes to the standard-defined input structures.

2.4.1 OSI Object Identifiers Must Include Decimal Values

All input object identifier component values must include decimal values. That is, the algorithm cannot accept object identifiers which specify ASN.1 labels alone without corresponding decimal values provided somewhere in the input. This assumption is consistent with the recommendations in reference **GDMO** Section 7.4, and the rules in reference **ASN.1**, Annex B to Annex D.

2.4.2 IMPORTed Data Types Must be Included with Input

IMPORTed data types from other OM packages do not need to be redefined, since they have already been defined previously within another OM package. For example, IMPORTs from CMIP (for example, **ObjectInstance**, **CMISFilter**) do not need redefinition, since they already exist in the XMP CMIS OM package. However, all data types IMPORTed from other ASN.1 modules must be available as input to the algorithm. To accomplish this, all imported OM packages must be identified in the OM-PACKAGE template as shown in Section 2.1 on page 5. Another alternative would be to require a list of imported ASN.1 modules. The mechanism used to access the listed OM package(s) and associated ASN.1 module(s) is implementation-dependent.

2.4.3 Uniqueness of OM Package Template Input

The algorithm assumes that values provided in the OM package template are unique (that is, information model name, abbreviated name, unique starting value, the OSI Object Identifier to be assigned to the generated OM package). For example, the unique abbreviated name and OM package object identifiers are used to prevent possible C Header File **#define** collisions when more than one OM package is used in the same workspace. A mechanism or registration process is needed to ensure uniqueness of this input.

2.5 Possible Input Extensions

The following input specification extensions were considered for use by the algorithm defined in this document, but are considered beyond its current scope.

2.5.1 Alternative Input Formats

This algorithm might be extended to accept other input formats such as Internet Structure of Management Information (SMI), OSI/Network Management Forum Object Specification Framework (OSF), or ANSI T1M1 pre-GDMO MACROs. However, for the most part, work is already underway to translate these alternative formats into ISO GDMO format (for example, see reference **NMFStrategy for Migration to GDMO**). For this reason, only ISO GDMO templates have been included in the current algorithm. Tools that might be developed to automate this algorithm might include preprocessors which convert alternative input formats into GDMO.

2.5.2 Practical Implementation Restrictions

The implementation of this algorithm as a tool needs to consider the operating system, the hardware platform, and the programming language issues and restrictions, such as maximum name lengths. For example, ANSI C requires that the first 32 characters of a name are unique. Although the GDMO template labels cannot be modified or abbreviated once registered, a tool might accept additional inputs to direct the output of the names to be shorter in the C Header File to accommodate the name length restriction. These types of implementation restrictions are beyond the scope of this document, but must be considered when implementing the algorithm.

Algorithm Output Specification

This chapter provides a summary description of the OM package output generated by the translation algorithm. It concludes with a description of output constraints that are placed on the algorithm, and a brief discussion of possible output extensions which might also be useful, but are beyond the current scope of this algorithm.

Figure 3-1 illustrates and expands on the details of the outputs of the algorithm and identifies the target environment with which the outputs are associated. The OM Package Definitions are in the form of OM definition tables and are intended for documentation purposes and the XMP API module. The C Header File is in the form of a typical C include file and is intended for the application module. The Encoding Definitions identify the encoding information required for the XMP API to perform the OM/BER encoding/decoding of the information model.

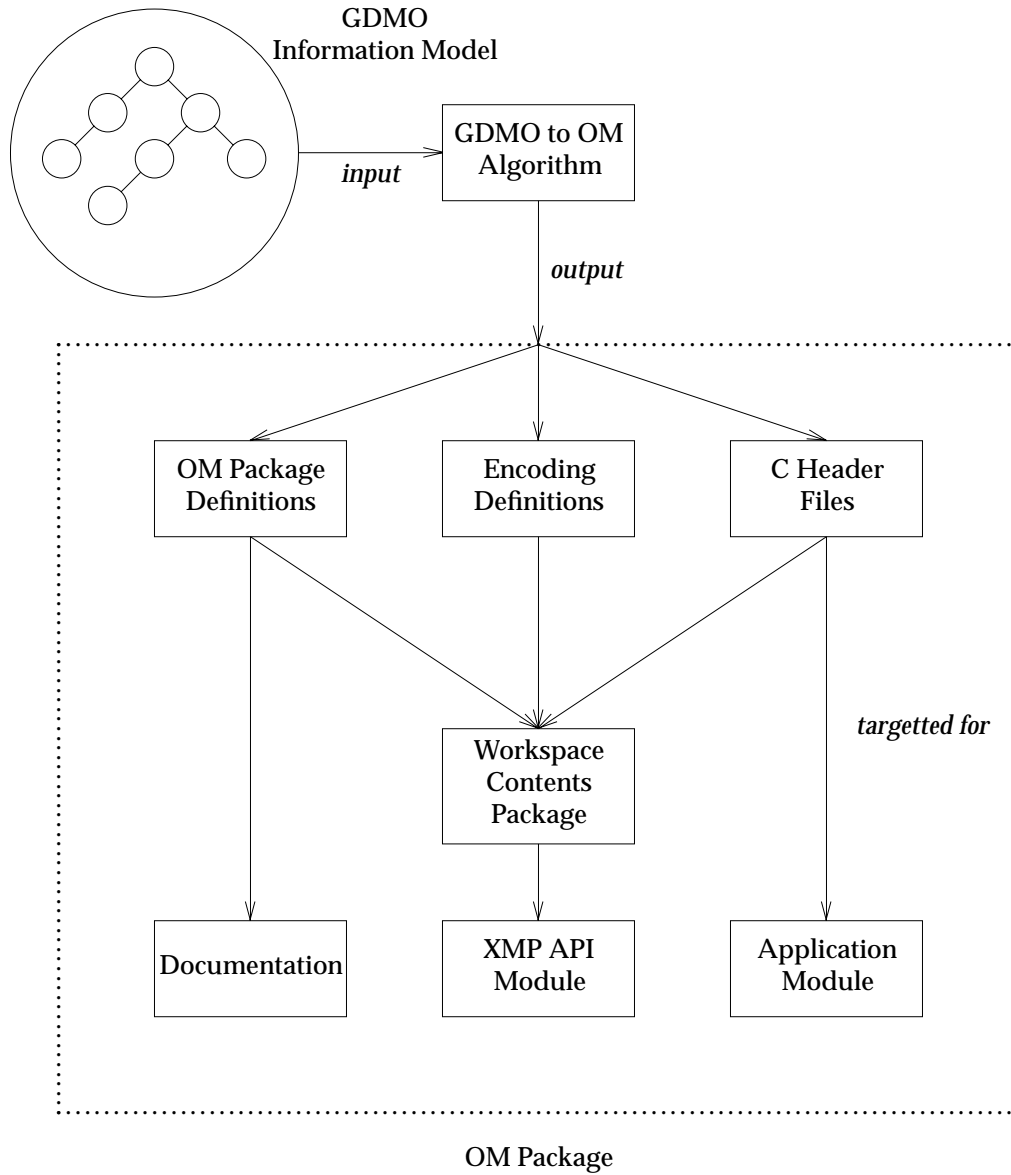


Figure 3-1 GDMO to OM Algorithm Architecture Details

3.1 OM Package Definitions

A significant portion of the output generated by the translation algorithm consists of the *OM Package Definitions*. In this context, the OM Package Definitions represent the input information model in the form of OM package definition tables. The OM Package Definitions are represented in OM notations defined in the XOM API.

The OM Package Definitions are divided into five areas: an OM Package Object Identifier, Object Identifier Tables, Information Syntax Tables, OM Attribute Tables and Value Lists.

Throughout this section, examples are provided from a sample DMI information model; refer to Appendix A for the complete OM Package Definitions generated from this sample input.

3.1.1 OM Package Object Identifier

The OM Package Object Identifier provides the OSI Object Identifier assigned to the entire OM package that represents the information model. The following example illustrates the OM Package Object Identifier for the sample DMI information model:

The SMI_PART2 (DMI) package is assigned the OSI Object Identifier { 1 2 826 0 1050 1 4 }.

3.1.2 Object Identifier Tables

The Object Identifier Tables list the OSI Object Identifiers assigned to OSI Managed Object Classes, Attributes (and Attribute Groups), Notifications, Actions and Parameters. For example, a portion of the Object Identifier Table for DMI Managed Object Classes looks like:

Object Class	Object Identifier
O-Discriminator	{ 2 9 3 2 3 3 }
O-System	{ 2 9 3 2 3 13 }
O-Top	{ 2 9 3 2 3 14 }

Tables containing other object identifiers for Attributes (and Attribute Groups), Notifications, Actions and Parameters have a similar format. The object identifiers which are derived from an information model are used in the following OM attributes and OM classes in the XMP API.

Object Identifier for:	Used in OM Attribute:	Of OM Class:
Object Class	<i>global-Form</i>	Object-Class
Attribute	<i>global-Form</i> <i>Naming-Attribute-Id</i>	Attribute-Id AVA
Attribute Group	<i>global-Form</i>	Attribute-Id
Notification	<i>global-Form</i>	Event-Type-Id
Action	<i>global-Form</i>	Action-Type-Id
Parameter	<i>error-Id</i>	Specific-Error-Info

Table 3-1 Object Identifier Usage

Other object identifiers associated with an information model, such as name bindings or conditional packages, do not affect the API directly and so do not appear in the OM Package Definition.

3.1.3 Information Syntax Tables

The Information Syntax Tables provide a list of OM classes which represent the syntax of an Attribute, Notification, Action or Parameter. These tables resolve the *any* syntaxes which appear in the XMP API. For example, a portion of the Information Syntax Table for DMI Notifications looks like:

Notification Type	Information Syntax	Reply Syntax
N-Attribute-Value-Change	Object(Attribute-Value-Change-Info)	-
N-State-Change	Object(State-Change-Info)	-

Tables containing information and reply syntaxes for Notifications and Actions have a similar format. Tables containing Attribute or Parameter Syntaxes look like:

Attribute Type	Attribute Syntax
A-Administrative-State	Enum(Administrative-State)
A-Attribute-Identifier-List	Object(Attribute-Identifier-List)
A-Attribute-Value-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)
A-Correlated-Notifications	Object(Setof-Correlated-Notifications)

The following table summarises the information syntaxes which are derived from an information model, and are used in the OM classes and OM attributes in the XMP API which contain the *any* syntax.

OM Class	OM Attribute	Syntax Defined By:
Attribute	attribute-Value	attribute-Id
AVA	naming-Attribute-Value	naming-Attribute-Id
Modification	attribute-Value	attribute-Id
CMIS-Event-Report-Argument	event-Info	event-Type
Event-Reply	event-Reply-Info	event-Type
Action-Info	action-Info-Arg	action-Type
Action-Reply	action-Reply-Info	action-Type
Specific-Error-Info	error-Info	error-Id

Table 3-2 OM Attributes with Syntax “ANY”

3.1.4 OM Attribute Tables

The OM Attribute Tables provide a list of OM attributes which represent the content of each OM class. For each OM attribute, the attribute's name, syntax, length and number of values are provided¹. The following example defines the OM class **Setof-Attribute-Value-Change-Definition** and the contained OM class **Attribute-Value-Change-Definition**.

Setof-Attribute-Value-Change-Definition

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-Value-Change-Definition	Object(Attribute-Value-Change-Definition)	-	0-more

OM Attributes of Setof-Attribute-Value-Change-Definition

Attribute-Value-Change-Definition

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-Id	Object(Attribute-Id)	-	1
old-Attribute-Value	any	-	0-1
new-Attribute-Value	any	-	1

OM Attributes of Attribute-Value-Change-Definition

In this example, the OM class Attribute-Id is IMPORTed from, and therefore defined by, the XMP API.

1. The algorithm specified by this document does not include OM attribute initial values for content-specific information because they are not required by the XMP API itself. GDMO attribute DEFAULT values are an application concern; the actual values are never conveyed by the CMIP protocol or service. ASN.1 DEFAULT values are not currently included in a table output, but could be handled by adding a "Value Initially" column to OM Class tables as illustrated in XOM (see reference **XOM API**).

When the OM class represents information defined as an ASN.1 CHOICE, this is indicated by additional text above the table, for example:

System-Id

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one of the following attributes must be present in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
name	String(Graphic)	-	0-1
number	Integer	-	0-1
nothing	Null	-	0-1

OM Attributes of System-Id

3.1.5 Value Lists

Value Lists enumerate the values which can be assigned to OM attributes of type **Enum**, **Integer** (when representing an ASN.1 named integer list), **String(Object-Identifier)** and **String(Bit)**. The following example defines the Value List for the Operational-State.

Operational-State

Value List for Operational-State:

operational-State.

Its value is one of:

- **disabled**
- **enabled**.

3.2 C Header File

Another significant portion of the output generated by the translation algorithm consists of the *C Header File*. The C Header File output contains generic C-language **#define** statements representing the information model which can be used by an application implementation using the XMP API.

The C Header File outputs can be divided into three areas:

- the OM Package Object Identifier
- the Object Identifier Definitions
- the OM Class Object Identifier, Name Constants and Value List Definitions.

The name of the C Header File is also generated during translation.

3.2.1 OM Package Object Identifier

The OM Package Object Identifier provides the OSI Object Identifier assigned to the entire OM package that represents the information model. The following example illustrates the OM Package Object Identifier for the DMI information model:

```
#define OMP_O_SMI_PART2  "\x2A\x86\x3A\x00\x88\x1A\x06\x04"
```

The OSI Object Identifier is represented in the form of a hexadecimal encoded string in the **#define** statement above.

An intermediate object identifier macro is defined, based on the OM Package Object Identifier:

```
/* Intermediate Object Identifier Macro */
mpP_dmi(X) (OMP_O_SMI_PART2##X)
```

3.2.2 Object Identifier Definitions

Each GDMO template has a unique object identifier associated with it. OSI Object Identifiers are defined for OSI Managed Object Classes, Attributes (and Attribute Groups), Notifications, Actions and Parameters. Note that other OSI object identifiers which may be defined as part of the input (for example, object identifiers assigned to ASN.1 modules or abbreviated superior registration arcs) do not appear in the translated output, since they are not used by the XMP API. Object identifiers are defined as shown in the following example:

```
/* Managed Object Classes */
#define OMP_O_DMI_O_SYSTEM  "\x59\x03\x02\x03\x0D"

/* Attributes */
#define OMP_O_DMI_A_DISCRIMINATOR_ID  "\x59\x03\x02\x07\x01"

/* Attribute Groups */
#define OMP_O_DMI_A_STATE  "\x59\x03\x02\x08\x01"

/* Notifications */
#define OMP_O_DMI_N_ATTRIBUTE_VALUE_CHANGE  "\x59\x03\x02\x0A\x01"

/* Parameters */
#define OMP_O_DMI_S_MISCELLANEOUS_ERROR  "\x59\x03\x02\x05\x01"
```

3.2.3 OM Class Constants

The OM Class Object Identifiers, Name Constants and Value List Definitions pertain to all data types in the ASN.1 syntax module. An OM class applies to all complex abstract syntaxes in the ASN.1 syntax module. Each OM class has associated with it an object identifier. The Name Constants apply to the OM classes, the OM attributes and the enumeration constants. The Value List Definitions apply to the enumerations, named integer lists and bit strings defined within the ASN.1 syntax module.

- Object Identifiers for OM classes

All complex data types in the ASN.1 syntax module are mapped to OM classes. Each OM class identifies itself using an object identifier. For example:

```
#define OMP_O_C_DMI_ATTRIBUTE_VALUE_CHANGE_INFO mpP_dmi(\xD8\x2C)
```

- OM Attribute Name Constants

All complex data types in the ASN.1 syntax module that are mapped to OM classes are composed of OM attributes. Each OM attribute has a name and a value that identifies itself. For example:

```
#define DMI_ATTRIBUTE_ID 11308
```

- Value List Constants

All data types in the ASN.1 syntax module that are ENUMERATED are mapped to Value List Constants which provide a list of enumeration names and values. Similar Value List Constants are also provided for data types in the ASN.1 module that are named INTEGER lists, BIT STRINGs or (in specified cases) OBJECT IDENTIFIERS. For example, for the DMI AdministrativeState data type, the Enumeration Value List Constants are as follows:

```
#define DMI_ADMINISTRATIVE_STATE_LOCKED          0
#define DMI_ADMINISTRATIVE_STATE_UNLOCKED       1
#define DMI_ADMINISTRATIVE_STATE_SHUTTING_DOWN  2
```

3.3 Encoding Definitions

The last significant portion of the output generated by the translation algorithm consists of the *Encoding Definitions*. In this context, the Encoding Definitions describe the type of encoding information required for the XMP API to perform the OM/BER encoding/decoding of the information model for the application.

The Encoding Definitions can be divided into two areas:

- the OM Class Encoding Definitions
- the OM Attribute Encoding Definitions.

3.3.1 OM Class Encoding Definitions

The OM class pertains to the complex abstract syntaxes of the data types in the ASN.1 syntax module. The encoding information that is needed to perform the encoding/decoding functionality is the OM class type. The different OM class types that exist are SEQUENCE, SET, SEQUENCE OF, SET OF, CHOICE or NONE. This information identifies which type of OM object the particular OM class definition represents.

The following example illustrates the OM Class Encoding Definition generated for the OM class **AttributeValueChangeInfo** OM object:

OM Class	OM Class Type
Attribute-Value-Change-Info	SEQUENCE

3.3.2 OM Attribute Encoding Definitions

The OM Attribute pertains to the complex abstract syntaxes of the data types in the ASN.1 syntax module. The encoding information that is needed to perform the encoding/decoding functionality is the OM attribute ASN.1 tagging information. The following list identifies the types of tagging information needed:

- implicit or Explicit tags
- tag type: Universal, Application, Context or Private
- tag value if not universal.

For simplicity, all the OM attributes of an OM class should follow each OM class description so that the API can associate the correct OM attributes with the corresponding OM class².

2. There are other methods/approaches that might be used to associate OM attributes with OM classes. The approach used here is for illustration purposes and for simplicity.

The following example illustrates the OM Attribute Encoding Definition generated for the OM class **AttributeValueChangeInfo** OM object:

OM Attribute	Implicit/Explicit	Tag Type	Tag Value
<i>source-Indicator</i>	IMPLICIT	UNIVERSAL	-
<i>attribute-Identifier-List</i>	IMPLICIT	CONTEXT	1
<i>attribute-Value-Change-Definition</i>	IMPLICIT	UNIVERSAL	-
<i>notification-Identifier</i>	IMPLICIT	UNIVERSAL	-
<i>correlated-Notifications</i>	IMPLICIT	CONTEXT	2
<i>additional-Text</i>	IMPLICIT	UNIVERSAL	-
<i>additional-Information</i>	IMPLICIT	CONTEXT	3

3.4 Workspace Contents Package Output Format

The output format for the XMP API module incorporates the outputs described in Section 3.1 on page 15, Section 3.2 on page 19 and Section 3.3 on page 21, into a single format that can be loaded into the XMP API software module. This is a generic format that any XMP API implementation can use to load the content-specific OM packages into the API workspace.

For example:

```

SMI-PART2 {1.2.826.0.1050.6.4} BEGIN
  IMPORTS CMIS-Filter (2021),
  Attribute (2006),
  Base-Managed-Object-Id (2011),
  Attribute-Id (2008),
  Object-Instance (2051),
  Object-Class (2050),
  Event-Type-Id (2037)
  FROM OMP_O_MP_CMIS_PKG {1.2.826.0.1050.6.2};
  AE-Title (1005),
  DS-DN (1017)
  FROM OMP_O_MP_COMMON_PKG {1.2.826.0.1050.6.1};
  ATTRIBUTE VALUES
  Administrative-State {2.9.3.2.7.31} SYNTAX ENUMERATION;
  :
  NOTIFICATIONS
  Attribute-Value-Change {2.9.3.2.10.1} SYNTAX Attribute-Value-Change-Info;
  :
  OBJECT DESCRIPTION
  Additional-Information (11301) SETOF CONCRETE DERIVED FROM OM_OBJECT
  ATTRIBUTE BEGIN
  management-Extension (11340) SYNTAX Management-Extension MULTIVALUED;
  END
  :
END

```

3.5 Output Constraints

The following constraint is imposed on output specifications generated by the algorithm defined in this document.

3.5.1 Reuse of Existing Package Definitions

Any data types IMPORTed from other OM packages, like the XMP OM packages, are not translated by the algorithm to OM objects because these data types are already defined in their own OM package. The ASN.1 Module IMPORTED FROM clause identifies other ASN.1 Modules from which data types are IMPORTed. ASN.1 type or value references in GDMO templates may also be prefixed by an external ASN.1 Module name from which the data type is IMPORTed. To allow for reuse of existing OM Package Definitions, the OM Package Template List-of-Imported-Packages clause must identify the OM Packages which contain the existing OM objects representing the IMPORTed data types.

3.6 Possible Output Extensions

The following output extensions were considered for generation by the algorithm defined in this document, but are not currently addressed.

3.6.1 No Abstract OM Classes Generated

The algorithm generates OM classes from ASN.1 syntaxes which all represent concrete OM classes. XOM specifies that an abstract class may be defined as a superclass in order to share attributes between classes, or to ensure that the class hierarchy is convenient for the interface definition. All data types in ASN.1 syntax modules specify the actual syntax that is carried in protocol for content-specific information, and are thus all concrete classes. Therefore, the algorithm does not create abstract OM classes as superclasses to the concrete OM classes generated from ASN.1 syntaxes. Also, since all OM classes generated by the algorithm are concrete, the parent object identifier is always OM-OBJECT. Isolating common characteristics to generate abstract OM classes is a subject for future study.

3.6.2 Updating Existing OM Packages

An existing OM package generated by this algorithm may require update to reflect a new version of the information model. In this case, syntax which has not changed must be IMPORTed from the existing OM package, and syntax which has changed or been added must be defined in a new OM package. The combination of the existing OM package and the new OM package therefore represent the new version of the information model. The starting value used to assign unique OM classes and OM attribute name constants in the new OM package must be larger than any defined in the existing OM package, to avoid collision in the package closure.

3.6.3 Single Intermediate Output Format

Another viewpoint that this algorithm can address is to generate an intermediate output format to represent all information required by both the application and the XMP API. Currently, this algorithm addresses the needs of the XMP API only, but could be extended to address the application's needs as well. This intermediate format could then be used as input to specific algorithms to generate the particular information needed by the application or the XMP API. For example, information in the GDMO templates which is useful for the application but not the API includes matching rules, default values, and the knowledge of which attributes, notifications and actions are contained within each GDMO package. This information is not included in the current algorithm output because it is not needed by the XMP API.

Detailed Translation Algorithm

This chapter defines the detailed translation algorithm for generating X/Open OM packages from ISO GDMO templates. The translation details are divided into eight sections:

- OM Package Template
- Managed Object Class Template
- Parameter Template
- Attribute Template
- Attribute Group Template
- Action Template
- Notification Template
- ASN.1 Module³.

Within each section, the Input Structure and the Output Structures are specified, a list of translation rules are provided which transform the Input Structure into the Output Structures, and an example is provided which illustrates how the translation rules are used.

This chapter concludes with a summary of the translation rules, including general naming conventions.

The casing rules used in this document conform to those used in the XOM API and the XMP API.

In this chapter:

- ***italicised-bold*** text represents content-specific information model data input used, and output generated, by the translation process. As shown by example, ***italicised-bold*** text is replaced by actual text which appears in the input information model and the outputs.
- UPPERCASE text represents ISO GDMO or ASN.1 keywords or reserved clauses which appear in the input information model.
- Text enclosed within square brackets [] is optional and may be omitted from the input or output.

3. This does not imply that the algorithm is required to perform the translation process in this order. In fact, this order cannot guarantee most efficient processing of input information. For example, OM syntax might best be derived once and applied in all the rules which reference it.

4.1 OM Package Template

This section provides the translation of the name of a package and its associated object identifier into an OM package. Note that the information provided by this template is also used in a number of other rules in the other sections.

4.1.1 Input Structure

Information-Model-Name OM-PACKAGE
 ABBREVIATION **Abbreviated-Name**
 INITIAL ATTRIBUTE VALUE **Unique-Starting-Value**
 [IMPORTED OM-PACKAGES **List-of-Imported-Packages**]
 REFERENCE "**Textual-Reference-To-Input-Source**"
 ::= { **Object-Identifier** }

This input structure is defined by this document to contain the additional information required by the algorithm but not already available in GDMO templates or ASN.1 modules. All clauses contained in this input structure, except for the IMPORTED OM-PACKAGES clause, are mandatory. The IMPORTED OM-PACKAGES clause is optional and its absence specifies that no OM objects are imported. The clauses are defined as follows:

- **Information-Model-Name** contains a unique label representing the Information Model, to be used as the basis for the OM package name. The name must be a valid C language identifier. It must be a sequence of letters, digits, and underscores, and it must not begin with a digit.
- **Abbreviated-Name** contains a unique abbreviation for the Information Model, to be used as a prefix throughout the OM package output; maximum of 5 characters in length. The abbreviation must be a valid C language identifier. It must be a sequence of letters, digits, and underscores, and it must not begin with a digit.
- **Unique-Starting-Value** is a positive integer which will be used as the starting point for assigning OM class and OM attribute name constants; unique constants are required within the OM package closure.
- **List-of-Imported-Packages** is a list of existing OM packages which contain OM objects reused by this OM package as IMPORTed data types; each OM package is listed by its OSI Object Identifier, separated by commas.
- **Textual-Reference-To-Input-Source** is a brief, free-format, text statement, which provides a reference to the source document(s) in which the input GDMO templates are defined. If the OM package translates only a subset of the material specified by the referenced document, that subset should be clearly identified here using clause numbers or template labels. To include a double quote character as part of the text statement, use two occurrences of the double quote character.
- **Object-Identifier** is a unique registered OSI Object Identifier which is assigned to the new OM package to be generated by the algorithm.

See reference [ASN.1](#) for additional information regarding the format of OSI Object Identifiers.

4.1.2 Output Structures

OM Package Definitions:

The **Information-Model-Name** (**Abbreviated-Name**) Package is assigned the OSI Object Identifier { Dec-**Object-Identifier** }.

C Header File <omp_**Abbreviated-Name**.h>

```
#define OMP_O_Information-Model-Name "Hex-Object-Identifier"
/* Intermediate Object Identifier Macro */
mpP_Abbreviated-Name(X) (OMP_O_Information-Model-Name#X)
```

Workspace Contents Package Output:

```
<OM-Package-Name> (<Abbreviated-Name>) {object-identifier} BEGIN
IMPORTS
[ <object-label> (<class-num>) [, <object-label> (<class-num>)]*
  FROM <package-label> {object-identifier};
[ <object-label> (<class-num>) [, <object-label> (<class-num>)]*
  FROM <package-label> {object-identifier};]* ]
```

4.1.3 Translation Rules

1. Convert the **Object-Identifier** input into its decimal value representation, Dec-**Object-Identifier**.

Any ASN.1 label used in the **Object-Identifier** is replaced by its actual value. This is done by locating the referenced label in the associated ASN.1 syntax module, and substituting the defined value for the label. After all labels have been substituted, any remaining text in the **Object-Identifier** is removed so that only decimal values remain. Decimal values must have been provided for all components of the **Object-Identifier**, either instead of, or in addition, to text. This rule is consistent with the recommendations in reference **GDMO**, Section 7.4, and the rules in reference **ASN.1**, Annex B to Annex D.

2. Replace the **Information-Model-Name**, the **Abbreviated-Name** and the **Object-Identifier** in the OM Package Definition Structure with the actual text.

The decimal value representation of the **Object-Identifier** is used. The **Abbreviated-Name** can only be a maximum of 3 characters long.

3. Replace the **Abbreviated-Name**, **Information-Model-Name** and the **Object-Identifier** in the C Header File Structure with the actual text.

The **Abbreviated-Name** appears in the name of the C Header File, with the "omp_" prefix and the ".h" suffix. The **Information-Model-Name** appears in the #define statement, with the "OMP_O" prefix. The decimal value representation of the **Object-Identifier** is converted into its hexadecimal encoded string representation.

4. The **Abbreviated-Name**, **Unique-Starting-Value** and the **List-of-Imported-Packages** are used in Section 4.8.
5. Replace the **Abbreviated-Name** and **Information-Model-Name** in the Intermediate Object Identifier Macro with the actual text. The **Abbreviated-Name** should appear in lowercase.
6. The <OM-Package-Name>, <Abbreviated-Name> and its associated "object-identifier" are copied from the **Information-Model-Name**, **Abbreviated-Name** and the **Object-Identifier** from the OM Package template, respectively.

The IMPORTS clause <package-label> and <object-label> are derived from the List-of-Imported-Packages in the OM Package template and the IMPORTS clause(s) of the ASN.1

module(s). The <class-num> is the OM class number of the imported OM class, defined in the referenced OM Package. The Package label object-identifier is the Package registration number from where the OM class(es) is(are) imported. ASN.1 Primitive types that are IMPORTed by the ASN.1 module do not appear in the IMPORTS clause if the Workspace Contents Package. All OM classes which are IMPORTed from other OM Packages are listed here so that they may be referenced elsewhere in the Workspace Contents output without ambiguity as to which OM Package they belong to.

4.1.4 Example

For example, given the Input:

```

SMI_PART2 OM-PACKAGE
  ABBREVIATION DMI
  INITIAL ATTRIBUTE VALUE 11301
  IMPORTED OM-PACKAGES
  { iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) common(1) },
  { iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) cmis(2) }
  REFERENCE "This package translates all templates defined by Rec. X.721 | ISO/IEC 10165-2 : 1992"
  ::= { iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) dmi(4) }

```

the following OM Package Definition output is produced:

The **SMI_PART2 (DMI)** package is assigned the OSI Object Identifier { **1 2 826 0 1050 6 4**}.

the following C Header File <omp_dmi.h> output is produced:

```

#define OMP_O_SMI_PART2  "\x2A\x86\x3A\x00\x88\x1A\x06\x04"
/* Intermediate Object Identifier Macro */
mpP_dmi(X)  (OMP_O_SMI_PART2##X)

```

and the following Workspace Contents Package output is produced:

```

SMI-PART2 {1.2.826.0.1050.6.4} BEGIN
  IMPORTS CMIS-Filter (2021),
  Attribute (2006),
  Base-Managed-Object-Id (2011),
  Attribute-Id (2008),
  Object-Instance (2051),
  Object-Class (2050),
  Event-Type-Id (2037)
  FROM OMP_O_MP_CMIS_PKG {1.2.826.0.1050.6.2};
  AE-Title (1005),
  DS-DN (1017)
  FROM OMP_O_MP_COMMON_PKG {1.2.826.0.1050.6.1};

```


4.2 Managed Object Class Template

This section provides the translation of an ISO GDMO Managed Object Class template into an OM Package.

4.2.1 Input Structure

```
Managed-Object-Name MANAGED OBJECT CLASS
:
REGISTERED AS Object-Identifier;
```

4.2.2 Output Structures

OM Package Definition:

<u>Object Class</u>	<u>Object Identifier</u>
O- <i>Managed-Object-Name</i>	Dec- <i>Object-Identifier</i>

Object Identifier for *Abbreviated-Name* Object Classes

C Header File:

```
#define OMP_O_Abbreviated-Name_O_Managed-Object-Name "Hex-Object-Identifier"
```

4.2.3 Translation Rules

1. Convert the *Object-Identifier* input into its decimal value representation, Dec-*Object-Identifier*.

See Section 4.1.3 Step 1 for details.

2. Replace the *Abbreviated-Name*, the *Managed-Object-Name* and the *Object-Identifier* in the OM Package Definition Structure with the actual text from the MANAGED OBJECT CLASS clause and from the REGISTERED AS clause.

The *Abbreviated-Name* appears in the Object-Identifier Table heading. The *Managed-Object-Name* appears in the Object Class column of the table, with the "O-" prefix. The decimal value representation of the *Object-Identifier* is used.

3. Replace the *Managed-Object-Name* and the *Object-Identifier* in the C Header File Structure with the actual text from the MANAGED OBJECT CLASS clause and from the REGISTERED AS clause.

The *Managed-Object-Name* appears in the #define statement, with the "OMP_O_*Abbreviated-Name*_O_" prefix. The decimal value representation of the *Object-Identifier* is converted into its hexadecimal encoded string representation.

4.2.4 Example

For example, given the Input:

```

system MANAGED OBJECT CLASS
  DERIVED FROM top
  CHARACTERIZED BY
  systemPackage PACKAGE
  ATTRIBUTES
    systemId      GET,
    systemTitle   GET,
    operationalState  GET,
    usageState    GET;;;
  CONDITIONAL PACKAGES
    administrativeStatePackage PACKAGE
  ATTRIBUTES
    administrativeState  GET-REPLACE;
  REGISTERED AS { smi2Package 14 } ;
    PRESENT IF "an instance supports it.",
    supportedFeaturesPackage PACKAGE
  ATTRIBUTES
    supportedFeatures  GET-REPLACE ADD-REMOVE;
  REGISTERED AS { smi2Package 15 } ;
    PRESENT IF "an instance supports it.";
  REGISTERED AS { smi2Package 26 };

```

then the following OM Package Definition output is produced:

Object Class	Object Identifier
O-System	{ 2 9 3 2 3 13 }

Object Class for **DMI** Object Classes

and the following C Header File output is produced:

```
#define OMP_O_DMI_O_SYSTEM  "\x59\x03\x02\x03\x0D"
```

4.3 Parameter Template

This section provides the translation of an ISO GDMO Parameter template into an OM Package.

4.3.1 Input Structure

```
Parameter-Name PARAMETER
  CONTEXT Context-Type;
  WITH SYNTAX Module-Name.Type-Name;
  :
  REGISTERED AS Object-Identifier;
```

or

```
Parameter-Name PARAMETER
  CONTEXT Context-Type;
  ATTRIBUTE Attribute-Label;
  :
  ;
```

4.3.2 Output Structures

OM Package Definition:

Parameter	Object Identifier
<i>S-Parameter-Name</i>	Dec- <i>Object-Identifier</i>

Object Identifiers for *Abbreviated-Name* Parameters

Parameter Type	Parameter Syntax
<i>S-Parameter-Name</i>	<i>OMsyntax</i>

Abbreviated-Name Parameter Value Syntaxes

C Header File:

```
#define OMP_O_ Abbreviated-Name_S_Parameter-Name "Hex-Object-Identifier"
```

Workspace Contents Package output:

```
PARAMETERS
[ <parameter-label> {object-identifier}
  [ SYNTAX <syntax> ]; ]*
```

4.3.3 Translation Rules

1. Convert the *Object-Identifier* input into its decimal value representation, Dec-*Object-Identifier*.

See Section 4.1.3 Step 1 for details.

If the REGISTERED AS clause does not appear in the template, then the ATTRIBUTE clause references the label of an ATTRIBUTE template defined elsewhere and no new syntax is defined. In this case, the PARAMETER template appears only in the Attribute Value Syntax table output, and not in the Object Identifier table output or the C Header File output.

2. Replace the **Abbreviated-Name**, the **Parameter-Name** and the **Object-Identifier** in the OM Package Definition Structure - Object Identifier Table with the actual text from the PARAMETER clause and from the REGISTERED AS clause.

The **Abbreviated-Name** appears in the Object-Identifier Table heading. The **Parameter-Name** appears in the Parameter column in the table, with the "S-" prefix. The decimal value representation of the **Object-Identifier** is used.

3. Replace the **Parameter-Name** and the **Object-Identifier** in the C Header File Structure - Object Identifier Definition with the actual text from the PARAMETER clause from the REGISTERED AS clause.

The **Parameter-Name** appears in the #define statement, with the "OMP_O_**Abbreviated-Name_S_**" prefix. The decimal value representation of the **Object-Identifier** is converted into its hexadecimal encoded string representation.

4. Generate the **OMsyntax** for the Parameter.

The **OMsyntax** is actually generated from the ASN.1 module later in this algorithm; refer to Section 4.8.3, Step 2. The OM syntax of the parameter syntax is derived by referencing the **Type-Name** in the ASN.1 **Module-Name** in the WITH SYNTAX clause.

5. Replace the **Abbreviated-Name**, **Parameter-Name** and **OMsyntax** in the OM Package Definition Structure - Information Syntax Table with the actual text.

The **Abbreviated-Name** appears in the Information Syntax Table heading. The **Parameter-Name** appears in the Parameter Type column of the table, with the "S-" prefix. The **OMsyntax** appears in the Parameter Syntax column of the table, derived from Step 4.

6. Replace the <parameter-label>, {object-identifier}, and <syntax> in the Workspace Contents Package output with the actual text from the PARAMETER clause and the REGISTERED AS clause, and any syntax derived from Step 4.

4.3.4 Example

For example, given the Input:

```

miscellaneousError PARAMETER
    CONTEXT          SPECIFIC-ERROR
    WITH SYNTAX      Parameter-ASN1Module.MiscellaneousError;
    BEHAVIOUR         miscellaneousErrorBehaviour
    BEHAVIOUR DEFINED AS "When a processing error failure has occurred
                        and the error condition does not match any of the object's
                        defined specific error types, this value is used.";

REGISTERED AS { smi2Parameter 1 };

```

the following OM Package Definition output is produced:

<u>Parameter</u>	<u>Object Identifier</u>
S-Miscellaneous-Error	{ 2 9 3 2 5 1 }
Object Identifier for DMI Parameters	

<u>Parameter Type</u>	<u>Parameter Syntax</u>
S-Miscellaneous-Error	Null
DMI Parameter Value Syntaxes	

then the following C Header File output is produced:

```
#define OMP_O_DMI_S_MISCELLANEOUS_ERROR "\x59\x03\x02\x05\x01"
```

and the following Workspace Contents Package output is produced:

```
PARAMETERS  
Miscellaneous-Error {2.9.3.2.5.1} SYNTAX NULL;
```

4.4 Attribute Template

This section provides the translation of an ISO GDMO Attribute template into an OM Package.

4.4.1 Input Structure

```

Attribute-Name ATTRIBUTE
  WITH ATTRIBUTE SYNTAX Module-Name.Type-Name
  :
  [ REGISTERED AS Object-Identifier ];
or
Attribute-Name ATTRIBUTE
  DERIVED FROM Attribute-Label;
  :
  REGISTERED AS Object-Identifier;

```

4.4.2 Output Structures

OM Package Definition:

<u>Attribute</u>	<u>Object Identifier</u>
A- <i>Attribute-Name</i>	Dec- <i>Object-Identifier</i>

Object Identifiers for *Abbreviated-Name* Attributes

<u>Attribute Type</u>	<u>Attribute Syntax</u>
A- <i>Attribute-Name</i>	<i>OMsyntax</i>

Abbreviated-Name Attribute Value Syntaxes

C Header File:

```
#define OMP_O_ Abbreviated-Name_A_ Attribute-Name    "Hex-Object-Identifier"
```

Workspace Contents Package output:

```

ATTRIBUTE VALUES
[ <attribute-label> {object-identifier}
  SYNTAX <syntax>; ]*

```

4.4.3 Translation Rules

1. Convert the *Object-Identifier* input into its decimal value representation, Dec-*Object-Identifier*.

See Section 4.1.3 Step 1 for details.

If the REGISTERED AS clause does not appear in the template, then the ATTRIBUTE cannot be used without further definition (for example, another ATTRIBUTE template which DERIVES FROM this one). In this case, the unregistered ATTRIBUTE template appears only in the *Attribute Value Syntax* table output, and not in the *Object Identifier* table output, the C Header File output, or the Workspace Contents Package output.

2. Replace the *Abbreviated-Name*, the *Attribute-Name* and the *Object-Identifier* in the OM Package Definition Structure - Object Identifier Table with the actual text from the ATTRIBUTE clause and from the REGISTERED AS clause.

The **Abbreviated-Name** appears in the Object Identifier Table heading. The **Attribute-Name** appears in the Attribute column in the table, with the "A-" prefix. The decimal value representation of the **Object-Identifier** is used.

3. Replace the **Attribute-Name** and the **Object-Identifier** in the C Header File Structure - Object Identifier Definition from the ATTRIBUTE clause and from the REGISTERED AS clause.

The **Attribute-Name** appears in the #define statement, with the "OMP_O_Abbreviated-Name_A_" prefix. The decimal value representation of the **Object-Identifier** is converted into its hexadecimal encoded string representation.

4. Generate the **OMsyntax** for the Attribute.

The **OMsyntax** is actually generated from the ASN.1 module later in this algorithm; refer to Section 4.8.3, Step 2. The OM syntax of the attribute syntax is derived by referencing the **Type-Name** in the ASN.1 **Module-Name** in the WITH ATTRIBUTE SYNTAX clause. If the DERIVED FROM clause is used instead of the WITH ATTRIBUTE SYNTAX, then the syntax defined in the **Attribute-Label** template is referenced.

5. Replace the **Abbreviated-Name**, **Attribute-Name** and **OMsyntax** in the OM Package Definition Structure - Information Syntax Table with the actual text.

The **Abbreviated-Name** appears in the Information Syntax Table heading. The **Attribute-Name** appears in the Attribute Type column of the table, with the "A-" prefix. The **OMsyntax** appears in the Attribute Syntax column of the table, derived from Step 4.

6. Replace the <attribute-label>, {object-identifier}, and <syntax> with the actual text from the ATTRIBUTE clause and the REGISTERED AS clause, and the syntax derived from Step 4.

4.4.4 Example

For example, given the Input:

```
discriminatorId ATTRIBUTE
  WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.SimpleNameType;
  MATCHES FOR EQUALITY, SUBSTRINGS, ORDERING;
  BEHAVIOUR rDNIDBehaviour;
  REGISTERED AS { smi2AttributeID 1};
```

the following OM Package Definition output is produced:

Attribute	Object Identifier
A-Discriminator-Id	{ 2 9 3 2 7 1 }

Object Identifiers for **DMI** Attributes

Attribute Type	Attribute Syntax
A-Discriminator-Id	Object(Simple-Name-Type)

DMI Attribute Value Syntaxes

then the following C Header File output is produced:

```
#define OMP_O_DMI_A_ DISCRIMINATOR_ID    "\x59\x03\x02\x07\x01"
```

and the following Workspace Package Contents output:

ATTRIBUTE VALUES

Discriminator-Id {2.9.3.2.7.1} SYNTAX Simple-Name-Type;

4.5 Attribute Group Template

This section provides the translation of an ISO GDMO Attribute Group template into an OM Package.

4.5.1 Input Structure

```
Attribute-Group-Name ATTRIBUTE GROUP
:
:
REGISTERED AS Object-Identifier;
```

4.5.2 Output Structures

OM Package Definition:

Attribute Group	Object Identifier
<i>A-Attribute-Group-Name</i>	<i>Dec-Object-Identifier</i>

Object Identifiers for *Abbreviated-Name* Attribute Groups

C Header File:

```
#define OMP_O_Abbreviated-Name_A_Attribute-Group-Name "Hex-Object-Identifier"
```

4.5.3 Translation Rules

1. Convert the *Object-Identifier* input into its decimal value representation, *Dec-Object-Identifier*.

See Section 4.1.3 Step 1 for details.

2. Replace the *Abbreviated-Name* and the *Attribute-Group-Name* and the *Object-Identifier* in the OM Package Definition Structure with the actual text from the ATTRIBUTE GROUP clause and from the REGISTERED AS clause.

The *Abbreviated-Name* appears in the Object Identifier Table heading. The *Attribute-Group-Name* appears in the Attribute Group column in the table, with the "A-" prefix. The decimal value representation of the *Object-Identifier* is used.

3. Replace the *Attribute-Group-Name* and the *Object-Identifier* in the C Header File Structure from the ATTRIBUTE GROUP clause and from the REGISTERED AS clause.

The *Attribute-Group-Name* appears in the #define statement, with the "OMP_O_*Abbreviated-Name*_A_" prefix. The decimal value representation of the *Object-Identifier* is converted into its hexadecimal encoded string representation.

4.5.4 Example

For example, given the Input:

```
state ATTRIBUTE GROUP
DESCRIPTION "This is defined as an empty group. The elements of
this group are composed of state attributes in the managed object.
The state attributes may include those specified in CCITT Rec. X.731
| ISO/IEC 10164-2 and others that are specific to the managed object class.";
REGISTERED AS { smi2AttributeGroup 1 };
```

then the following OM Package Definition output is produced:

<u>Attribute Group</u>	<u>Object-Identifier</u>
A- <i>State</i>	{ 2 9 3 2 8 1 }

Object Identifiers for *DMI* Attribute Groups

and the following C Header File output is produced:

```
#define OMP_O_DMI_A_STATE    "\x59\x03\x02\x08\x01"
```

4.6 Action Template

This section provides the translation of an ISO GDMO Action template into an OM Package.

4.6.1 Input Structure

```

Action-Name ACTION
:
  [ WITH INFORMATION SYNTAX Module-Name.Type-Name; ]
  [ WITH REPLY SYNTAX Module-Name.Type-Name; ]
REGISTERED AS Object-Identifier;

```

4.6.2 Output Structures

OM Package Definition:

Action	Object Identifier
T- Action-Name	{ Dec- Object-Identifier }

Object Identifiers for **Abbreviated-Name** Actions

Action Type	Information Syntax	Reply Syntax
T- Action-Name	OMsyntax or "-"	OMsyntax or "-"

Abbreviated-Name Action Information Syntaxes

C Header File:

```
#define OMP_O_ Abbreviated-Name_T_ Action-Name "Hex-Object-Identifier"
```

Workspace Contents Package output:

```

ACTIONS
[ <action-label> {object-identifier}
  [ SYNTAX <syntax> ] [ REPLY SYNTAX <syntax> ]; ]*

```

4.6.3 Translation Rules

1. Convert the **Object-Identifier** input into its decimal value representation, Dec-**Object-Identifier**.

See Section 4.1.3 Step 1 for details.

2. Replace the **Abbreviated-Name**, the **Action-Name** and the **Object-Identifier** in the OM Package Definition Structure - Object Identifier Table with the actual text from the ACTION clause and from the REGISTERED AS clause.

The **Abbreviated-Name** appears in the Object Identifier Table heading. The **Action-Name** appears in the Action column in the table, with the "T-" prefix. The decimal value representation of the **Object-Identifier** is used.

3. Replace the **Action-Name** and the **Object-Identifier** in the C Header File Structure - Object Identifier Definition from the ACTION clause and from the REGISTERED AS clause.

The **Action-Name** appears in the #define statement, with the "OMP_O_ **Abbreviated-Name**_T_" prefix. The decimal value representation of the **Object-Identifier** is converted into its hexadecimal encoded string representation.

4. Generate the **OMsyntaxes** for the Action Information and Reply.

The **OMsyntax** is actually generated from the ASN.1 module later in this algorithm; refer to Section 4.8.3, Step 2. The OM syntaxes of the action information and reply syntaxes are derived by referencing the **Type-Name** in the ASN.1 **Module-Name** in the WITH INFORMATION SYNTAX and the WITH REPLY SYNTAX clauses, respectively.

5. Replace the **Abbreviated-Name**, **Action-Name** and **OMsyntax** in the OM Package Definition Structure - Information Syntax Table with the actual text.

The **Abbreviated-Name** appears in the Information Syntax Table heading The **Action-Name** appears in the Action Type column of the table, with the "T-" prefix. The **OMsyntax** appears in the Information and Reply Syntax columns of the table. Each OM syntax is derived from Step 4. If the WITH INFORMATION SYNTAX or WITH REPLY SYNTAX clause is not defined in the input template, then a dash "-" appears in the corresponding column of the output table.

6. Replace the <action-label>, {object-identifier}, and <syntax>es in the Workspace Contents PACKage output with the actual text from the ACTION clause and the REGISTERED AS clause, and any information and reply syntaxes derived from Step 4.

4.6.4 Example

There are no ACTION templates defined by the sample input information model in reference DMI. For illustration, consider the following fictitious Activate ACTION (which does NOT appear in Appendix A):

```
Activate ACTION
  MODE CONFIRMED;
  WITH REPLY SYNTAX Action-ASN1Module.ActivateReply;
  REGISTERED AS { smi2Action 1 };
```

the following OM Package Definition output is produced:

Object Class	Object Identifier
T-Activate	{ 2 9 3 4 9 1 }

Object Identifiers for **DMI** Actions

Action Type	Information Syntax	Reply Syntax
T-Activate	-	Object(Activate-Reply)

DMI Action Information Syntaxes

the following C Header File output is produced:

```
#define OMP_O_DMI_T_ACTIVATE    "\x59\x03\x04\x09\x01"
```

and the following Workspace Contents Package output is produced:

```
ACTIONS
Activate {2.9.3.4.9.1} REPLY SYNTAX Activate-Reply;
```

4.7 Notification Template

This section provides the translation of an ISO GDMO Notification template into an OM Package.

4.7.1 Input Structure

```

Notification-Name NOTIFICATION
:
[ WITH INFORMATION SYNTAX Module-Name.Type-Name;
:
]
[ WITH REPLY SYNTAX Module-Name.Type-Name; ]
REGISTERED AS { Object-Identifier };

```

4.7.2 Output Structures

OM Package Definition:

Notification	Object Identifier
N-Notification-Name	{ Dec-Object-Identifier }

Object Identifiers for *Abbreviated-Name* Notifications

Notification Type	Information Syntax	Reply Syntax
N-Notification-Name	OMsyntax or "-"	OMsyntax or "-"

Abbreviate-Name Notification Information Syntaxes

C Header File:

```
#define OMP_O_Abbreviated-Name_N_Notification-Name "Hex-Object-Identifier"
```

Workspace Contents Package output:

```

NOTIFICATIONS
[ <notification-label> {object-identifier}
[ SYNTAX <syntax> ] [ REPLY SYNTAX <syntax> ]; ]*

```

4.7.3 Translation Rules

1. Convert the *Object-Identifier* input into its decimal value representation, Dec-*Object-Identifier*.

See Section 4.1.3 Step 1 for details.

2. Replace the *Abbreviated-Name*, the *Notification-Name* and the *Object-Identifier* in the OM Package Definition Structure - Object Identifier Table with the actual text from the NOTIFICATION clause and from the REGISTERED AS clause.

The *Abbreviated-Name* appears in the Object Identifier Table heading. The *Notification-Name* appears in the Notification column in the table, with the "N-" prefix. The decimal value representation of the *Object-Identifier* is used.

3. Replace the *Notification-Name* and the *Object-Identifier* in the C Header File Structure - Object Identifier Definition with the actual text from the NOTIFICATION clause and from the REGISTERED AS clause.

The **Notification-Name** appears in the #define statement, with the "N-" prefix. The decimal value representation of the **Object-Identifier** is converted into its hexadecimal encoded string representation.

4. Generate the **OMsyntaxes** for the Notification Information and Reply.

The **OMsyntax** is actually generated from the ASN.1 module later in this algorithm; refer to Section 4.8.3, Step 2. The OM syntaxes of the notification information and reply syntaxes are derived by referencing the **Type-Name** in the ASN.1 **Module-Name** in the WITH INFORMATION SYNTAX and the WITH REPLY SYNTAX clauses, respectively.

5. Replace the **Abbreviated-Name**, **Notification-Name** and **OMsyntax** in the OM Package Definition Structure - Information Syntax Table with the actual text.

The **Abbreviated-Name** appears in the Information Syntax Table heading. The **Notification-Name** appears in the Notification Type column of the table, with the "N-" prefix. The **OMsyntax** appears in the Information and Reply Syntax columns of the table. Each OM syntax is derived from Step 4. If the WITH INFORMATION SYNTAX or WITH REPLY SYNTAX clause is not defined in the input template, then a dash "-" appears in the corresponding column of the output table.

6. Replace the <notification-label>, {object-identifier}, and <syntax>es in the Workspace Contents Package output with the actual text from the NOTIFICATION clause, and the REGISTERED AS clause, and any information and reply syntaxes derived from Step 4.

4.7.4 Example

For example, given the Input:

```
attributeValueChange NOTIFICATION
BEHAVIOUR attributeValueChangeBehaviour;
WITH INFORMATION SYNTAX
Notification-ASN1Module.AttributeValueChangeInfo
AND ATTRIBUTE IDS
sourceIndicator          sourceIndicator,
attributeIdentifierList  attributeIdentifierList,
attributeValueChangeDefinition  attributeValueChangeDefinition,
notificationIdentifier  notificationIdentifier,
correlatedNotifications  correlatedNotifications,
additionalText          additionalText,
additionalInformation    additionalInformation;
REGISTERED AS { smi2Notification 1 };
```

the following OM Package Definition output is produced:

Notification	Object Identifier
N-Attribute-Value-Change	{ 2 9 3 4 10 1 }

Object Identifiers for **DMI** Notifications

Notification Type	Information Syntax	Reply Syntax
N-Attribute-Value-Change	Object(Attribute-Value-Change-Info)	-

DMI Notification Information Syntaxes

the following C Header File output is produced:

```
#define OMP_O_DMI_N_ATTRIBUTE_VALUE_CHANGE    "\x59\x03\x02\x0A\x01"
```

and the following Workspace Contents Package output is produced:

```
NOTIFICATIONS  
Attribute-Value-Change {2.9.3.2.10.1} SYNTAX Attribute-Value-Change-Info;
```

4.8 ASN.1 Module

This section provides the translation of an ASN.1 Module into an OM Package. Refer to reference **ASN.1** for details of the ASN.1 syntax types.

4.8.1 Input Structure

```

Module-Name { Object-Identifier }
DEFINITIONS Tag-Type ::=
BEGIN
:
IMPORTS
:
FROM Import-Module { Object-Identifier }

Type-Name ::= <<< ASN.1 simple data type definitions >>>
  <<< if enumerated type: Enumerated-Name (Enumerated-Value) >>>
  <<< if named integer type: Integer-Name (Integer-Value) >>>
  <<< if bit string type: Bit-Name (Bit-Value) >>>
OID-Name Type-Name ::= OID-Value
:
Type-Name ::= <<< ASN.1 constructed data type definitions >>> {
  [ASN1-Identifier] [Optional-Tag] ASN1-Type [OPTIONAL],
  :
}
:
END

```

where **Tag-Type** is either **IMPLICIT TAGS** or **EXPLICIT TAGS** or empty.
 If the **ASN1-Identifier** is absent, a "pseudo-identifier"
 is generated; refer to Section 4.8.3, Step 4.

4.8.2 Output Structures

OM Package Definition:

Type-Name

An instance of this OM class has the OM attributes of its superclass - Object - and additionally the OM attributes listed below. (Exactly one of the following attributes must be present in an instance of this OM class.)

OM Attribute	Value Syntax	Value Length	Value Number
ASN1-Identifier	OMsyntax	valueLength or "-"	valueNumber

OM Attributes of a **Type-Name**

Type-Name

Value List for **Type-Name**

Type-Name. Its value is one of:

- **Enumerated-Name** or **Integer-Name** or **Bit-Name** or **OID-Name**
- :
- **Enumerated-Name** or **Integer-Name** or **Bit-Name** or **OID-Name**

C Header File:

```
#define OMP_O_C_Abbreviated-Name_Type-Name
        mpP_Abbreviated-Name("Hex-OMobjectId-Trailer")

#define Abbreviated-Name_ASN1-Identifier        value

#define Abbreviated-Name_Type-Name_Enumerated-Name Enumerated-Value

#define Abbreviated-Name_Type-Name_Integer-Name Integer-Value

#define Abbreviated-Name_Type-Name_Bit-Name      Bit-Value

#define Abbreviated-Name_Type-Name_OID-Name     OID-Value
```

Encoding Definition:

1. The OM class type: SEQUENCE, SET, SEQUENCE OF, SET OF, CHOICE or NONE.
2. The OM attribute ASN.1 tagging information:
 - implicit or explicit tags
 - tag type: Universal, Application, Context or Private
 - tag value if not universal.

Workspace Contents Package output:

```
OBJECT DESCRIPTION
[<OM-object-description-template>]*
```

4.8.3 Translation Rules

1. Ignore most ASN.1 value definitions.

The values assigned to ASN.1 Object Identifier labels are resolved in the GDMO templates rules stated in previous sections. The values assigned to ENUMERATED, named INTEGER lists, OBJECT IDENTIFIERS and BIT STRING types are resolved in Step 8 below. Default ASN.1 values are not relevant to the algorithm. This set of rules therefore concentrates on translation of ASN.1 type definitions, and ignores all other ASN.1 value definitions.

2. Map the ASN.1 data type syntax to OM syntax.

Each ASN.1 data type syntax is mapped to an OM syntax. Each data type can be referenced via the WITH SYNTAX clause in the Attribute Template, Parameter Template, Notification Template or the Action Template, or via a constructed data type within the ASN.1 module or within another ASN.1 module. If the ASN.1 syntax is a type reference, the type reference assignment is checked to determine the mapping. Refer to the reference **XOM** for translation rules which map OM syntaxes to ASN.1 types.

The translation process starts with the ASN.1 Type-Name referenced by the WITH SYNTAX clause of the GDMO template. In the ASN.1 module, this Type-Name is assigned to ("::=") one of the valid ASN.1 syntaxes shown in Table 4-1. The ASN.1 'Selection Type' is treated simply as the type selected by the field name, i.e., the Type-Name is the field selected type. The rules in Table 4-2 are applied to the Type-Name based on the syntax given in Table 4-1. If the Type-Name is a Primitive, Constructed-2, or Imported-2, the translation is complete. Otherwise, Table 4-2 is used recursively to translate nested syntaxes until all translations are complete. When iterating through Table 4-2, Type-Name1 refers to the Type-Name being translated, and Type-Name2 refers to the "next level" Type-Name (for example, Type-Name1 ::= Type-Name2).

This translation process is shown in Figure 4-1.

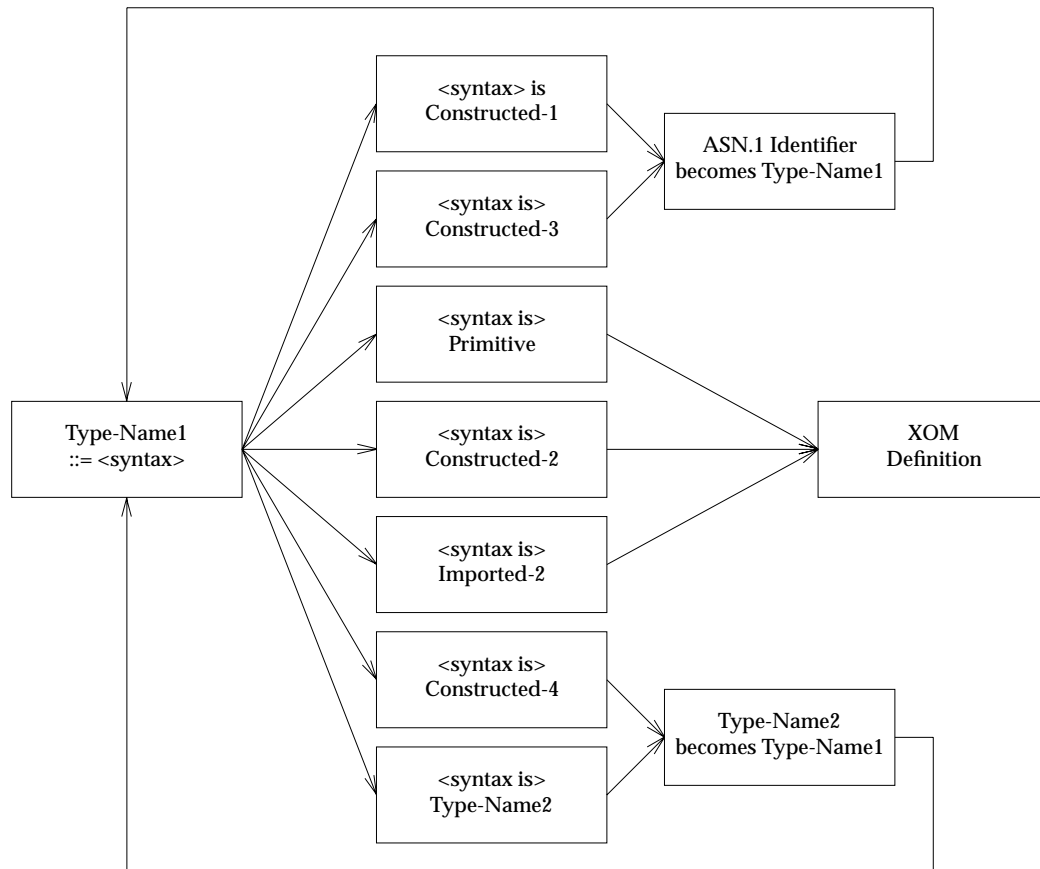


Figure 4-1 Overall Translation Process Flow

Two additional rules apply which are not addressed in the XOM API. The valid combinations of ASN.1 syntax type definitions to which the two rules apply are summarised in Table 4-1.

Table 4-1 Valid ASN.1 Syntax Type Definitions

Syntax Type	Meaning
Primitive	Boolean, Enumeration, Integer, Null, String(*), Any, or ASN.1 REAL.
Constructed-1	Choice, Set or Sequence.
Constructed-2	Set-of/Sequence-of Primitive.
Constructed-3 ⁵	Set-of/Sequence-of Constructed-1. Note that all other constructed combinations are not valid.
Constructed-4	Set-of/Sequence-of Type-Name.
Type-Name	The name assigned to the ASN.1 data type definition. The Type-Name can either be defined within the ASN.1 module or IMPORTed from another ASN.1 module.
Imported-1	IMPORTed from another ASN.1 module within the OM package.
Imported-2	IMPORTed from another ASN.1 module NOT within the OM package. Note that all other IMPORTed types from OM packages not previously defined are not allowed, because they violate the OM package closure rule. If the IMPORTed label is not represented as an OM class in an existing OM package, then: <ul style="list-style-type: none"> • If the label corresponds to a primitive type, the ASN.1 type definition is treated as a Primitive to be translated as part of the current OM package. • If the label corresponds to a constructed type, the entire exporting ASN.1 module may be processed to generate the OM class (translated into a separate OM package).
With-Components	Type-Name(WITH COMPONENTS {..}). Treated just like Type-Name, WITH COMPONENTS is ignored.

⁵ Set-of / Sequence-of any other Constructed (-2/3/4) is not addressed here, but might be considered as Constructed-4, where the other Constructed is handled as though it were a Type-Name. For example, SET OF SEQUENCE OF INTEGER can be handled as SET OF AType, where AType ::= SEQUENCE OF INTEGER.

Table 4-2 Valid OM Class and OM Attribute Name Constants

Type-Name1 has Syntax Type	OM Class and OM Attribute Name Constant Rules ⁶
Primitive	Type-Name1 is not used
Constructed-1	One OM object: OM Class Name - Type-Name1 OM Attribute Name - ASN-Identifiers within the constructed type (attributes which are components of a CHOICE class each have the valueNumber 0-1).
Constructed-2	One OM object: OM Class Name - Type-Name1 OM Attribute Name - Type-Name1

Type-Name1 has Syntax Type	OM Class and OM Attribute Name Constant Rules ⁶
	(valueNumber is 0-more).
Constructed-3	Two OM object: (1) OM Class Name - Setof-Type-Name1 OM Attribute Name - Type-Name1 (valueNumber is 0-more). (2) OM Class Name - Type-Name1 OM Attribute Name - ASN1-Identifiers within the constructed type
Constructed-4	One OM object: OM Class Name - Type-Name1 OM Attribute Name - Type-Name2 (valueNumber is 0-more). Replace Type-Name1 with Type-Name2 and apply this table's process again.
Type-Name2	Type-Name1 is not used. Replace Type-Name1 with Type-Name2 and apply this table's process again.
Imported-2	One OM object: OM Class Name - Type-Name1 (class defined in another OM package)

⁶ When a duplicate OM Class Name is generated by following these rules, duplicates are assigned suffixes for uniqueness, starting with "-1" for the first duplicate encountered in the ASN.1 module and incrementing by one for each duplicate encountered thereafter. Duplicate OM Class Names derived from multiple ASN.1 modules are also resolved in this manner.

- Determine the **valueLength** and **valueNumber** for each OM attribute.

The **valueLength** is derived if an ASN.1 value range or a SIZE construct appears in the Input Structure data type definition. The **valueNumber** for each OM attribute is determined based on the ASN.1 syntax. Unless an exception is noted in Table 4-2, the default rule that applies is that the **valueNumber** is "0-1" if the ASN.1 keyword OPTIONAL or DEFAULT appears in the input module definition, otherwise the **valueNumber** is "1".

- Replace the **Type-Name**, **ASN-Identifier**, **OMsyntax**, **valueLength** and **valueNumber** in the OM Package Definition Structure - OM Attribute Tables with the actual text.

An OM Attribute Table must be generated for every OM class defined by the package, except for those OM classes already defined in the XMP API (for example, Object-Class, Object-Instance) or another OM package in the **List-of-Imported-Packages**. Each table contains standard introductory text as shown previously. Tables which represent ASN.1 CHOICE data types also include the sentence:

"Exactly one of the following attributes must be present in an instance of this OM class."

The **Type-Name** appears in the table heading. The **ASN1-Identifier** appears in the OM Attribute column of the table. The **OMsyntax** appears in the Value Syntax column of the table, derived from Step 2. The **valueLength** appears in the Value Length column of the table, derived from Step 3. If no such construct appears, a "-" is placed in this column of the table. The **valueNumber** appears in the Value Number column of the table, derived

from Step 3.

If the **ASN1-Identifier** is absent from the input, a *pseudo-identifier* is generated to serve as the **ASN1-Identifier**. In this case, the **ASN1-Identifier** is assumed to be **Type-Name-ASN1-Type-#**, where # is an integer value assigned sequentially within the type, starting from "1". For example,

```
Sample ::= CHOICE { INTEGER, BOOLEAN }
```

would become

```
Sample ::= CHOICE { sample-INTEGGER-1 INTEGER, sample-BOOLEAN-2 BOOLEAN }.
```

Attributes appear within the table in the order in which they appear in the ASN.1 module being translated.

5. Replace the **Type-Name**, **Enumerated-Name**, **Integer-Name**, **OID-Name** and **Bit-Name** in the OM Package Definition Structure - Value Lists, with the actual text.

A Value List must appear for every OM attribute of type **Enum**, named **Integer** list, **String(object-identifier)**, or **String(Bit)** defined by the package. The **Type-Name** appears in the list heading and introductory sentence. Each **Enumerated-Name**, **Integer-Name**, **OID-Name** or **Bit-Name** appears as a dashed list item, the ASN.1 label broken apart into individual words and lower-cased for readability.

6. Replace the **Abbreviated-Name** and the **Type-Name** for the constructed types in the C Header File Structure - OM Class OID and Name Constants with the actual text, and define object identifiers for them.

The OM Class Name Constants are generated by concatenating the **Abbreviated-Name** and the **Type-Name**. Including the **Abbreviated-Name** within the name constant ensures uniqueness across OM packages. Once all OM Classes have been identified and alphabetised in ascending order, unique numbers are assigned sequentially, starting with **Unique-Starting-Value** and incrementing by one. Each **Hex-OMObjectId-Trailer** is replaced by a hexadecimal encoded string containing the unique number assigned to the OM class. **mpP_Abbreviated-Name** is replaced by the actual Intermediate Object Identifier Macro defined in Section 4.1.3 Step 5.

7. Replace the **Abbreviated-Name** and the **ASN1-Identifier** of the constructed type in the C Header File Structure - OM Attribute Name Constants with the actual text, and define values for them.

The OM Attribute Name Constants are generated by concatenating the **Abbreviated-Name** and the **ASN1-Identifier**. If the **ASN1-Identifier** is absent from the input, a *pseudo-identifier* is generated to serve as the **ASN1-Identifier** as described in Step 4. Once all the OM attributes have been identified, duplicate OM Attribute Name Constants have been removed, and the OM attribute list has been alphabetised in ascending order, the decimal values are generated. Unique values are assigned sequentially, starting with **Unique-Starting-Value** and incrementing by one, to the alphabetised list of OM attributes.

8. Replace the **Abbreviated-Name**, the **Type-Name** and the **Enumerated-Name** and **Enumerated-Value** or **Integer-Name** and **Integer-Value** or **Bit-Name** and **Bit-Value** or **OID-Name** and **OID-Value** in the C Header File Structure - Value List Constants with the actual values.

The Value List Constants are generated by concatenating the **Abbreviated-Name** and the **Type-Name** with either the **Enumerated-Name** of ENUMERATED data types, or the **Integer-Name** of named INTEGER list data types, or the **Bit-Name** of BIT STRING types, or the **OID-Name** of OBJECT IDENTIFIERS. Constant Values are assigned using the

Enumerated-Values, **Integer-Values**, **Bit-Values**, or **OID-Values** (in hex) specified in the data type definition. The resulting Value List Constants are guaranteed to be unique by this derivation rule.

Note that Value List Contents should be generated only for those OBJECT IDENTIFIERS which can be used as the value of a generated OM-attribute. Other OBJECT IDENTIFIERS which may appear in the input ASN.1 module (for example, the object identifier assigned to the ASN.1 module itself) should not be translated into Value List Constants.

9. Define the OM class type for all the **Type-Name** constructed types in the Encoding Definition Structure - OM Class Encoding Definitions.

The OM class type is determined based on the following rules:

- If the ASN.1 data type is a SET OF, SEQUENCE OF, SET, SEQUENCE, or CHOICE, then the corresponding OM class type appears in the output.
 - Otherwise (if the ASN.1 data type is anything else), then the keyword "NONE" appears in the OM class type output.
 - The NONE OM class type applies to those OM classes which type reference an IMPORTed data type from another OM Package.
10. Define the OM attribute tagging for all OM attributes in the Encoding Definition Structure - OM Attribute Encoding Definitions.

For simplicity, all the OM attributes of an OM class should follow each OM class description so that the API can associate the correct OM attribute with the corresponding OM class. Attributes appear within the table in the order in which they appear in the ASN.1 module being translated.

The OM attribute ASN.1 tagging information is determined based on the following rules:

- The **Tag-Type** in the DEFINITIONS clause and the **Optional-Tag** in the ASN.1 syntax determine whether an IMPLICIT or EXPLICIT tag is used, as follows:

DEFINITIONS	Optional-Tag	Tag Used
IMPLICIT TAGS	Not specified IMPLICIT EXPLICIT	IMPLICIT IMPLICIT EXPLICIT
EXPLICIT TAGS	Not specified IMPLICIT EXPLICIT	EXPLICIT IMPLICIT EXPLICIT
Not specified	Not specified IMPLICIT EXPLICIT	EXPLICIT IMPLICIT EXPLICIT

An exception occurs when the **ASN1-Type** is either an ANY type or a CHOICE type. In these cases, the tag used shall always be EXPLICIT.

- The tag type is determined on the **Optional-Tag** in the ASN.1 syntax. The tag type is UNIVERSAL if **Optional-Tag** does not exist. The tag type is APPLICATION if the keyword APPLICATION appears in the **Optional-Tag**. The tag type is PRIVATE if the keyword PRIVATE appears in the **Optional-Tag**. Otherwise, the tag type is CONTEXT.
- The tag value is derived from **Optional-Tag** if **Optional-Tag** exists.

11. Replace the <OM-object-description-template> in the Workspace Contents Package output with the actual text for each OM class generated by the preceding Steps, as further described in Section 4.9.

4.8.4 Example

For example, given the Input:

```

Attribute-ASN1Module { joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1 }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS everything
IMPORTS
    CMISFilter, Attribute, BaseManagedObjectId, AttributeId, ObjectInstance, ObjectClass, EventTypeId
FROM CMIP-1 { joint-iso-ccitt ms(9) cmip(1) version1(1) protocol(3) }
:
smi2AttributeID OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) smi(3) part2(2) attribute(7) }
smi2AttributeGroup OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) smi(3) part2(2) attributeGroup(8) }
-- default value definitions
defaultDiscriminatorConstruct DiscriminatorConstruct ::= and : {}
-- supporting productions
AdministrativeState ::= ENUMERATED { locked(0), unlocked(1), shuttingDown(2) }
AttributeValueChangeDefinition ::= SET OF SEQUENCE {
    attributeID AttributeId,
    oldAttributeValue [1] ANY DEFINED BY attributeID OPTIONAL,
    newAttributeValue [2] ANY DEFINED BY attributeID }
:
DiscriminatorConstruct ::= CMISFilter
:
SimpleNameType ::= CHOICE {
    number INTEGER,
    string GraphicString }
:
END

```

the following OM Package Definition output is produced:

Attribute-Value-Change-Definition

An instance of this OM class has the OM attributes of its superclass - Object - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-Id	Object(Attribute-Id)	-	1
old-Attribute-Value	any	-	0-1
new-Attribute-Value	any	-	1

OM Attributes of *Attribute-Value-Change-Definition*

Setof-Attribute-Value-Change-Definition

An instance of this OM class has the OM attributes of its superclass - Object - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-Value-Change-Definition	Object(Attribute-Value-Change-Definition)	-	0-more

OM Attributes of *Setof-Attribute-Value-Change-Definition*

Simple-Name-Type

An instance of this OM class has the OM attributes of its superclass - Object - and additionally the OM attributes listed below. Exactly one of the following attributes must be present in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
number	Integer	-	0-1
string	String(Graphic)	-	0-1

OM Attributes of *Simple-Name-Type*

Administrative-State

Value List for *Administrative-State*

administrative-State. Its value is one of:

- *locked*
- *unlocked*
- *shutting down*.

The following C Header File output is produced:

```

/* OM Class Object Identifiers */
:
#define OMP_O_C_DMI_ATTR_VALUE_CHANGE_DEFINITIONN      mpP_dmi(\xD8\x2B)
:
#define OMP_O_C_DMI_SETOF_ATTR_VALUE_CHANGE_DEFINITION  mpP_dmi(\xD8\x49)
:
#define OMP_O_C_DMI_SIMPLE_NAME_TYPE                  mpP_dmi(\xD8\x51)
:

/* Attribute Name Constants */

#define DMI_ATTRIBUTE_ID          11308
:
#define DMI_NEW_ATTRIBUTE_VALUE   11348
:
#define DMI_NUMBER                11356
:
#define DMI_OLD_ATTRIBUTE_VALUE   11364
:
#define DMI_STRING                11390

```

```

/* Value List Constants */

#define DMI_ADMINISTRATIVE_STATE_LOCKED      0
#define DMI_ADMINISTRATIVE_STATE_UNLOCKED   1
#define DMI_ADMINISTRATIVE_STATE_SHUTTING_DOWN 2
:

```

The following Encoding Definition output is produced:

OM Class	OM Class Type		
Attribute-Value-Change-Definition	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
Attribute-Id	IMPLICIT	UNIVERSAL	-
Old-Attribute-Value	EXPLICIT	CONTEXT	1
New-Attribute-Value	EXPLICIT	CONTEXT	2

OM Class	OM Class Type		
Setof-Attribute-Value-Change-Definition	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
Attribute-Value-Change-Definition	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Simple-Name-Type	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
Number	IMPLICIT	UNIVERSAL	-
String	IMPLICIT	UNIVERSAL	-

The following Workspace Contents Package output is produced:

```

OBJECT DESCRIPTION
Attribute-Value-Change-Definition (11307) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
  ATTRIBUTE BEGIN
    attribute-Id (11308) SYNTAX Attribute-Id MANDATORY;
    old-Attribute-Value (11364) SYNTAX ANYDEFINEDBY attribute-Id OPTIONAL WITH TAG [1];
    new-Attribute-Value (11348) SYNTAX ANYDEFINEDBY attribute-Id MANDATORY
      WITH TAG [2];
  END
Setof-Attribute-Value-Change-Definition (11337) SETOF CONCRETE DERIVED FROM OM_OBJECT
  ATTRIBUTE BEGIN
    attribute-Value-Change-Definition (11311) SYNTAX Attribute-Value-Change-Definition
      MULTIVALUED;
  END
Simple-Name-Type (11345) CHOICE CONCRETE DERIVED FROM OM_OBJECT
  ATTRIBUTE BEGIN
    number (11356) SYNTAX INTEGER OPTIONAL;
    string (11391) SYNTAX GRAPHICSTRING OPTIONAL;
  END

```

4.9 Workspace Contents Package Output Translation

This section describes the translation of input provided previously into a single output that can be used to load the contents package into the XMP API workspace.

4.9.1 Input Structure

This output is generated from input already provided in previous sections, as described in the translation rules.

4.9.2 Output Structure

The output format is described below as a set of production rules. The UPPERCASE text represents keywords in this output structure.

1. OM PACKAGE TEMPLATE

```

<OM-Package-Name> ( <Abbreviated-Name> ) {object-identifier} BEGIN
  IMPORTS
  [ <object-label> ( <class-num> ) [, <object-label> ( <class-num> ) ]*
    FROM <package-label> {object-identifier};
  [ <object-label> ( <class-num> ) [, <object-label> ( <class-num> ) ]*
    FROM <package-label> {object-identifier};]* ]
  ATTRIBUTE VALUES
  [<OM-attribute-values-template>]*
  NOTIFICATIONS
  [<OM-notification-template>]*
  ACTIONS
  [<OM-action-template>]*
  PARAMETERS
  [<OM-parameter-template>]*
  OBJECT DESCRIPTION
  [<OM-object-description-template>]*
END

```

2. OM ATTRIBUTE VALUES TEMPLATE

```

<attribute-label> {object-identifier}
  SYNTAX <syntax>;

```

3. OM NOTIFICATION TEMPLATE

```

<notification-label> {object-identifier}
  [SYNTAX <syntax>]
  [REPLY SYNTAX <syntax>];

```

4. OM ACTION TEMPLATE

```

<action-label> {object-identifier}
  [SYNTAX <syntax>]
  [REPLY SYNTAX <syntax>];

```

5. OM PARAMETER TEMPLATE

```

<parameter-label> {object-identifier}
  SYNTAX <syntax>;

```

6. OM OBJECT DESCRIPTION TEMPLATE

```

<object-label> (<class-num>) [SEQUENCE | SET | SEQUENCEOF | SETOF | CHOICE]
                                         CONCRETE | ABSTRACT
DERIVED FROM OM_OBJECT | <class-name> {object-identifier}
ATTRIBUTE BEGIN
  <om-attribute-label> [<om-attribute-label>]*
END

```

Attributes appear within the table in the order in which they appear in the ASN.1 module being translated.

7. OM ATTRIBUTE TEMPLATE

```

<om-attribute-label> (<const-type>)
SYNTAX <syntax>
[LENGTH <length>]
MANDATORY | OPTIONAL | MULTIVALUED
[WITH TAG [<number> | APPLICATION <number> | PRIVATE <number>] [IMPLICIT];

```

8. Representation of <syntax>

```

<syntax> ::= <object-label> | <universal-label>
<universal-label> ::= BOOLEAN | INTEGER | BITSTRING | REAL |
OCTETSTRING | NULL | ANY | ANYDEFINEDBY | EXTERNAL |
OBJECTIDENTIFIER | GRAPHICSTRING |
NUMERICSTRING | PRINTABLESTRING |
TELETEXSTRING | VIDEOTEXSTRING |
VISIBLESTRING | IA5STRING |
GENERALSTRING | GENERALIZEDTIME |
UTCTIME | ENUMERATED

```

4.9.3 Translation Rules

1. The <OM-Package-Name>, <Abbreviated-Name> and its associated *object-identifier* are copied from the **Information-Model-Name**, **Abbreviated-Name** and the **Object-Identifier** from the OM Package template, respectively.
2. The <class-num> and the <const-type> are derived from the **Unique-Starting-Value** in the OM Package template.
3. All others are derived from the GDMO Templates and ASN.1 Syntax module translation rules previously described in Section 4.2 through Section 4.8. Note that attributes without **Object Identifiers** are not included in the OM Attribute Values Template.
4. The IMPORTS clause <package-label> and <object-label> are derived from the List-of-Imported-Packages in the OM Package template and the IMPORTS clause(s) of the ASN.1 module(s). The <class-num> is the OM class number of the imported OM class, defined in the referenced OM Package. The Package label object-identifier is the Package registration number from where the OM class(es) is(are) imported. ASN.1 Primitive types that are IMPORTed by the ASN.1 module do not appear in the IMPORTS clause of the Workspace Contents Package. All OM classes which are IMPORTed from other OM Packages are listed here so that they may be referenced elsewhere in the Workspace Contents output without ambiguity as to which OM Package they belong to.

5. The **valueNumber** generated in Section 4.8.3, Step 3 is used to determine the number of values keyword in the OM Attribute Template, as follows:

valueNumber	keyword
1	MANDATORY
0-1	OPTIONAL
0-more	MULTIVALUED

4.9.4 Example

Refer to Section 3.4 on page 23 for an example of the Workspace Contents Package Output.

4.10 Summary of the Translation Rules

The translation rules defined by the previous sections are summarised in three categories. The first category summarises the rules that apply to both the OM Package Definition and C Header Files. The second category summarises rules that apply to only the C Header Files. The third category summarises rules that apply to only the Encoding Definitions.

Figure 4-2 illustrates and expands on the details of the type of inputs and outputs of the algorithm.

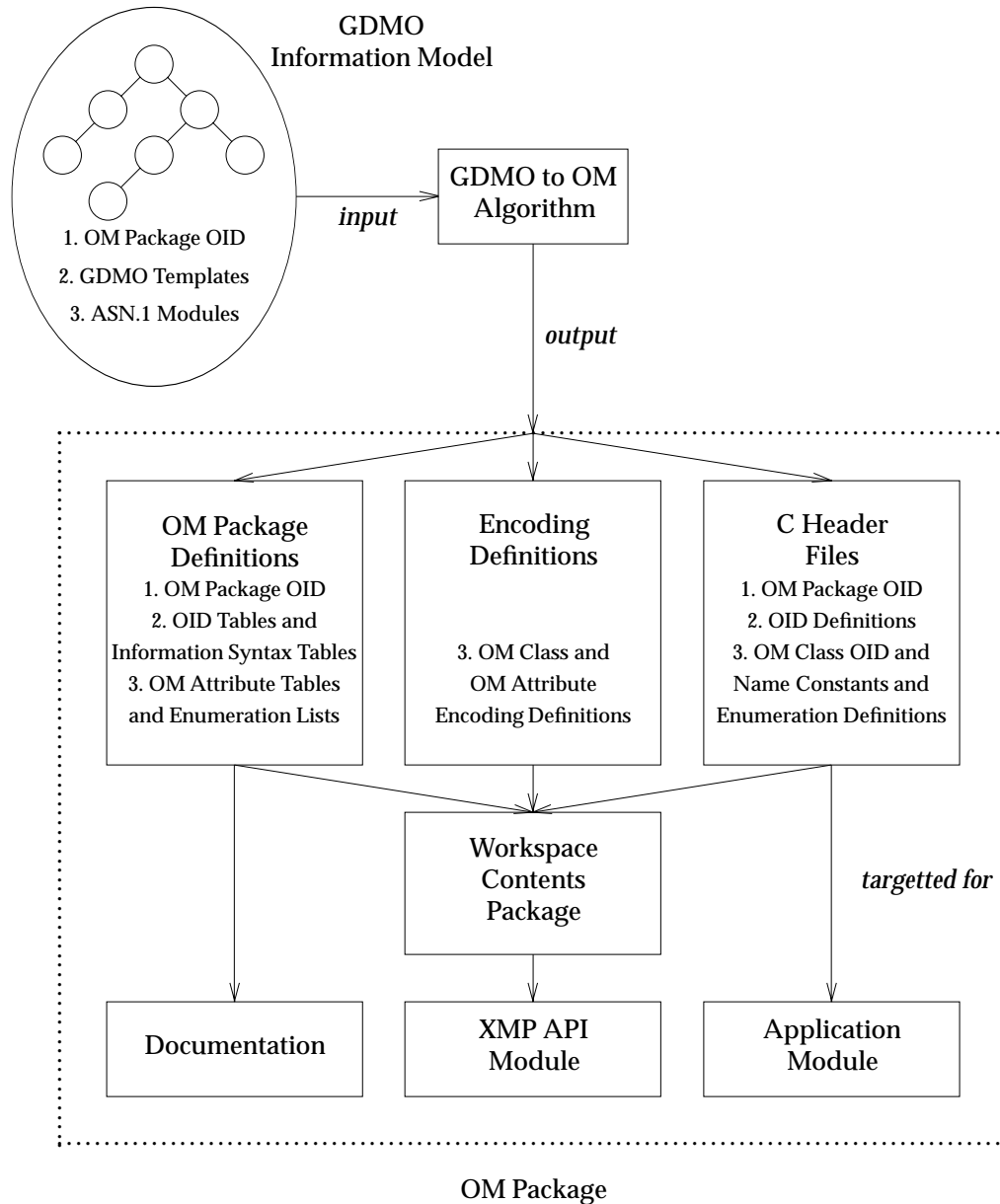


Figure 4-2 GDMO to OM Algorithm Inputs and Outputs

Figure 4-2 shows which inputs affect which outputs via the translation algorithm. The algorithm takes the OM Package OID input and generates an OM Package OID for the OM Package Definition and the C Header File. The algorithm takes the GDMO Templates input and generates OID tables and Information Syntax for the OM Package Definition and OID Definitions for the C Header File. The algorithm takes the ASN.1 modules input and generates the OM Attribute Tables and Value Lists for the OM Package Definition, the OM Class OID, Name Constants and Value List Constants, and OM Class and OM Attribute Syntaxes for the C Header File, and the OM Class and OM Attribute Encoding Definitions.

4.10.1 Rules for OM Package Definition and C Header Files

The following summary of translation rules apply to both the OM Package Definition and the C Header Files:

1. Naming Conventions - Casing Rules

The casing rules used in this document conform to those used in the reference **XOM** and the reference **XMP**, for both the OM Package Definition and the C Header File.

As described in reference **XMP**, ASN.1 labels are converted into OM class and attribute names by breaking them into individual words and inserting dashes between them. For example, the ASN.1 label *stateChangeDefinition* becomes *state-Change-Definition*. In general, upper-case characters and dashes (-) signal the start of a new word. Exceptions to this general rule include:

- acronyms composed entirely of upper-case characters - these are treated together as a single word (for example, CMIS-Filter)
- numeric characters - these are treated as lower-case and thus are considered part of the preceding word (for example, Time24).

2. Object Identifier Representation

The Object-Identifier input is converted into its decimal value representation by following the detailed rules specified in Section 4.1.3, Step 1. The decimal value representation of the **Object-Identifier** is used in the OM Package Definition. The decimal value representation of the **Object-Identifier** is converted into its hexadecimal encoded string representation in the C Header File.

3. ASN.1 types to OM Syntax Mapping

Each ASN.1 data type syntax is mapped to an OM syntax. Each data type can be referenced via the WITH SYNTAX clause in the Attribute Template, Parameter Template, Notification Template or the Action Template, or via a constructed data type within the ASN.1 module or within another ASN.1 module. If the ASN.1 syntax is a type reference, the type reference assignment is checked to determine the mapping. Any **Type-Name** that applies to a primitive type is not used. Refer to the reference **XOM** for translation rules which map OM syntaxes to ASN.1 types.

Two additional rules apply which are not addressed in the XOM API:

- If the **Type-Name** name appears directly in the IMPORT list FROM CMIP, Directory Services or another OM package in the **List-of-Imported-Packages**, then no OM syntaxes are generated, since they are already defined in the XMP API. Refer to Table 4-1 on page 49 for details of the rules.
- If the **Type-Name** is assigned to ("::=") any of the valid ASN.1 syntax types defined in Table 4-1 on page 49, then the rules in Table 4-2 on page 49 apply.

4.10.2 Rules for C Header Files Only

The following summary of translation rules apply to only the C Header Files:

1. C Naming Conventions - Prefixes

The C naming conventions used in this document conform to those specified in the reference **XOM** and the reference **XMP**. The OMP_ prefix is used to define the interface extension of the content-specific information in the C Header File.

2. Constants and OM Class and OM Attribute Value Generation

Constants for the OM class are created by concatenating the **Abbreviated-Name** with the **Type-Name**. The Constant for the OM attribute is created by concatenating the **Abbreviated-Name** with the **ASN1-Identifier**. Value List Constants are created by concatenating the **Abbreviated-Name** and the **Type-Name** with either the **Enumeration-Name**, the **Integer-Name**, the **Bit-Name**, or the **OID-Name**. These Constants ensure uniqueness across OM packages. Note that, for the sake of brevity, OM Class and attribute names appearing in other output formats do not include the **Abbreviated-Name** prefix. In all cases, lower case letters and dashes are converted to upper case letters and underscores, respectively, prior to alphabetising using the ASCII collating sequence.

The OM class object identifier is generated by concatenating the OM Package Object Identifier with a number, by using an intermediate object identifier macro. Numbers are assigned sequentially to the alphabetised list of OM classes, beginning with a **Unique-Starting-Value**, and concatenated to the OM Package Object Identifier to ensure uniqueness. OM attribute values are also assigned sequentially to the alphabetised list of all the OM attributes, after first removing any duplicate OM attributes.

Refer to Section 4.8.3, Steps 6, 7 and 8 for details of the rules.

4.10.3 Rules for Encoding Definitions Only

1. OM Class Type

The class type information is needed for all the OM classes to allow the API implementation to encode/decode the content-specific data to/from OM/BER for the application. The OM class type information is one of the following: SEQUENCE, SET, SEQUENCE OF, SET OF, CHOICE or NONE. Refer to Section 4.6.3 Step 9 for details of the rules.

2. OM Attribute ASN.1 Tagging Information

The tagging information is needed for all the OM attributes to allow the API implementation to encode/decode the content-specific data to/from OM/BER for the application. The ASN.1 tagging information consists of the following information:

- implicit or explicit tags
- the tag type: Universal, Application, Context or Private
- the tag value.

Refer to Section 4.8.3 Step 10 for details of the rules.

DMI Contents Package

This appendix describes the DMI Contents Package for use with X/Open's System Management: Management Protocol API (reference **XMP**). The Package had been generated by applying the GDMO-XOM translation algorithm to the GDMO definitions referenced below.

This Package is provided as an example of the application of the translation algorithm.

Note: In the case of any conflict between the definition of the algorithm, and the example output provided in this appendix, the algorithm is to be taken as authoritative.

Each Package specifies OM classes addressing the structure of the management information proper to notification types, attribute types, and action types which are used in the definition of managed object classes. Packages define the abstract syntaxes and semantics of the management information which is exchanged between the programs, that is, the contents of the notifications and operations.

This appendix sets out for the DMI Package the names for each of these items, and defines OM classes to represent those which are not represented directly by OM syntaxes. The values of attributes in the Management messages are not restricted to those discussed in this document, and new attribute types and syntaxes may be created at any time. Implementations are likely to add additional definitions. The Management Protocols API defines how the values of other syntaxes are represented in the interface.

The constants and OM classes defined in this specification are additional to those presented in the Interface Class Definitions included in the Management Protocols API since they are not essential to the working of the interface, but instead allow systems management functions to be utilised.

This DMI Package document contains all the definitions related to the management messages of Systems Management Functions specified in reference **SMF** parts 1 through 7, together with the definitions contained in reference **DMI**.

The DMI (Structure of Management Information - Part 2) package is assigned the OSI Object Identifier:

{iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) dmi(4)}

Note: OM modules derived from different levels of object definitions in GDMO are not necessarily interoperable. The identical level of the object definition should be verified to ensure interoperability. For example, ISO/IEC 10165-2 (DMI) is likely to be changed by an ISO Technical Corrigendum. This will cause the original DMI OM modules not to be fully interoperable with the new DMI OM modules reflecting the ISO Technical Corrigendum changes.

The constants which represent the OM classes and OM attributes in the C binding are defined in the `<omp_dmi.h>` header - see Appendix A.

In order to generate the DMI Package from the rules specified in the algorithm for translating GDMO to XOM objects, the following OM Package Template is defined:

SMI_PART2 OM-PACKAGE
ABBREVIATION DMI
INITIAL ATTRIBUTE VALUE 11301
IMPORTED OM-PACKAGES
{iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) common(1)},
{iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) cmis(2)}
REFERENCE "This package translates all templates defined by Rec. X.721 | ISO/IEC 10165-2 : 1992"
::= {iso(1) member-national-body(2) bsi(826) disc(0) xopen(1050) xmp-cae(6) dmi(4)}

A.1 Attributes and Event Types

This specification presents the attribute and event types restricted to those related to the following Systems Management Functions:

- Object Management Function
- State Management Function
- Attributes for Representing Relationships
- Alarm Reporting Function
- Event Report Control Function
- Log Control Function
- Security Alarm Reporting Function.

The form of each value of an attribute is determined by the attribute syntax associated with the attribute's type.

Each management notification is made up of an event type together with one event information value. The form of each value of an event type is determined by the event syntax associated with the event's type.

In the XMP interface, attributes appear as instances of OM class *Attribute*, with the attribute type represented as the value of the OM attribute *Attribute-Id* and the attribute value(s) represented as the value(s) of the OM attribute *Attribute-Value*. Each attribute type has an object-identifier, assigned in the Definition of Management Information standard (reference **DMI**), which is the value of the OM attribute *Attribute-Id*.

In the interface, events appear as a couple of information through various OM classes defined in the CMIS package of the XMP API, wherein the event type is represented as the value of the OM attribute *Event-Type-Id*, and the event information value is represented as the value(s) of the OM attribute *Event-Info* or *Event-Reply-Info*. Each event type has an object-identifier, assigned in the DMI document, which is the value of the OM attribute *Event-Type*.

External Definitions

The following OM classes, used in the DMI Package, are defined in XMP API, in the CMIS Management Service Package and in the Common Management Service Package.

- **AE-Title**
- **Attribute**
- **Attribute-Id**
- **CMIS-Filter**
- **DS-DN**
- **Event-Type-Id**
- **Object-Class**
- **Object-Instance.**

A.2 Object Identifiers for DMI

A.2.1 DMI OM Package Object Identifier

The SMI_PART2 (DMI) package is assigned the OSI Object Identifier { 1 2 826 0 1050 6 4 }.

A.2.2 Object Identifiers for DMI Object Classes

Object Class	Object Identifier
O-Alarm-Record	{ 2 9 3 2 3 1 }
O-Attribute-Value-Change-Record	{ 2 9 3 2 3 2 }
O-Discriminator	{ 2 9 3 2 3 3 }
O-Event-Forwarding-Discriminator	{ 2 9 3 2 3 4 }
O-Event-Log-Record	{ 2 9 3 2 3 5 }
O-Log	{ 2 9 3 2 3 6 }
O-Log-Record	{ 2 9 3 2 3 7 }
O-Object-Creation-Record	{ 2 9 3 2 3 8 }
O-Object-Deletion-Record	{ 2 9 3 2 3 9 }
O-Relationship-Change-Record	{ 2 9 3 2 3 10 }
O-Security-Alarm-Report-Record	{ 2 9 3 2 3 11 }
O-State-Change-Record	{ 2 9 3 2 3 12 }
O-System	{ 2 9 3 2 3 13 }
O-Top	{ 2 9 3 2 3 14 }

A.2.3 Object Identifiers for DMI Attributes

Attribute	Object Identifier
A-Active-Destination	{ 2 9 3 2 7 49 }
A-Additional-Information	{ 2 9 3 2 7 6 }
A-Additional-Text	{ 2 9 3 2 7 7 }
A-Administrative-State	{ 2 9 3 2 7 31 }
A-Alarm-Status	{ 2 9 3 2 7 32 }
A-Allomorphs	{ 2 9 3 2 7 50 }
A-Attribute-Identifier-List	{ 2 9 3 2 7 8 }
A-Attribute-List	{ 2 9 3 2 7 9 }
A-Attribute-Value-Change-Definition	{ 2 9 3 2 7 10 }
A-Availability-Status	{ 2 9 3 2 7 33 }
A-Backed-Up-Object	{ 2 9 3 2 7 41 }
A-Backed-Up-Status	{ 2 9 3 2 7 11 }
A-Back-Up-Destination-List	{ 2 9 3 2 7 51 }
A-Back-Up-Object	{ 2 9 3 2 7 40 }
A-Capacity-Alarm-Threshold	{ 2 9 3 2 7 52 }
A-Confirmed-Mode	{ 2 9 3 2 7 53 }
A-Control-Status	{ 2 9 3 2 7 34 }
A-Correlated-Notifications	{ 2 9 3 2 7 12 }
A-Corrupted-PD-Us-Received-Counter	{ 2 9 3 2 7 72 }
A-Corrupted-PD-Us-Received-Threshold	{ 2 9 3 2 7 89 }

A-Current-Log-Size	{ 2 9 3 2 7 5 4 }
A-Destination	{ 2 9 3 2 7 5 5 }
A-Discriminator-Construct	{ 2 9 3 2 7 5 6 }
A-Discriminator-Id	{ 2 9 3 2 7 1 }
A-Event-Time	{ 2 9 3 2 7 1 3 }
A-Event-Type	{ 2 9 3 2 7 1 4 }
A-Incoming-Connection-Reject-Error-Counter	{ 2 9 3 2 7 7 3 }
A-Incoming-Connection-Reject-Error-Threshold	{ 2 9 3 2 7 9 0 }
A-Incoming-Connection-Requests-Counter	{ 2 9 3 2 7 7 4 }
A-Incoming-Connection-Requests-Threshold	{ 2 9 3 2 7 9 1 }
A-Incoming-Disconnect-Counter	{ 2 9 3 2 7 7 5 }
A-Incoming-Disconnect-Error-Counter	{ 2 9 3 2 7 7 6 }
A-Incoming-Disconnect-Error-Threshold	{ 2 9 3 2 7 9 2 }
A-Incoming-Protocol-Error-Counter	{ 2 9 3 2 7 7 7 }
A-Incoming-Protocol-Error-Threshold	{ 2 9 3 2 7 9 3 }
A-Intervals-Of-Day	{ 2 9 3 2 7 5 7 }
A-Log-Full-Action	{ 2 9 3 2 7 5 8 }
A-Logging-Time	{ 2 9 3 2 7 5 9 }
A-Log-Id	{ 2 9 3 2 7 2 }
A-Log-Record-Id	{ 2 9 3 2 7 3 }
A-Managed-Object-Class	{ 2 9 3 2 7 6 0 }
A-Managed-Object-Instance	{ 2 9 3 2 7 6 1 }
A-Max-Log-Size	{ 2 9 3 2 7 6 2 }
A-Member	{ 2 9 3 2 7 4 2 }
A-Monitored-Attributes	{ 2 9 3 2 7 1 5 }
A-Name-Binding	{ 2 9 3 2 7 6 3 }
A-Notification-Identifier	{ 2 9 3 2 7 1 6 }
A-Number-Of-Records	{ 2 9 3 2 7 6 4 }
A-Object-Class	{ 2 9 3 2 7 6 5 }
A-Octets-Received-Counter	{ 2 9 3 2 7 7 8 }
A-Octets-Received-Threshold	{ 2 9 3 2 7 9 4 }
A-Octets-Retransmitted-Error-Counter	{ 2 9 3 2 7 7 9 }
A-Octets-Retransmitted-Threshold	{ 2 9 3 2 7 9 5 }
A-Octets-Sent-Counter	{ 2 9 3 2 7 8 0 }
A-Octets-Sent-Threshold	{ 2 9 3 2 7 9 6 }
A-Operational-State	{ 2 9 3 2 7 3 5 }
A-Outgoing-Connection-Reject-Error-Counter	{ 2 9 3 2 7 8 1 }
A-Outgoing-Connection-Reject-Error-Threshold	{ 2 9 3 2 7 9 7 }
A-Outgoing-Connection-Requests-Counter	{ 2 9 3 2 7 8 2 }
A-Outgoing-Disconnect-Counter	{ 2 9 3 2 7 8 3 }
A-Outgoing-Disconnect-Error-Counter	{ 2 9 3 2 7 8 4 }
A-Outgoing-Disconnect-Error-Threshold	{ 2 9 3 2 7 9 9 }
A-Outgoing-Protocol-Error-Counter	{ 2 9 3 2 7 8 5 }
A-Outgoing-Protocol-Error-Threshold	{ 2 9 3 2 7 1 0 0 }
A-Owner	{ 2 9 3 2 7 4 3 }
A-Packages	{ 2 9 3 2 7 6 6 }
A-PD-Us-Received-Counter	{ 2 9 3 2 7 8 6 }
A-PD-Us-Received-Threshold	{ 2 9 3 2 7 1 0 1 }
A-PD-Us-Retransmitted-Error-Counter	{ 2 9 3 2 7 8 7 }
A-PD-Us-Retransmitted-Error-Threshold	{ 2 9 3 2 7 1 0 2 }
A-PD-Us-Sent-Counter	{ 2 9 3 2 7 8 8 }
A-PD-Us-Sent-Threshold	{ 2 9 3 2 7 1 0 3 }
A-Peer	{ 2 9 3 2 7 4 4 }
A-Perceived-Severity	{ 2 9 3 2 7 1 7 }
A-Primary	{ 2 9 3 2 7 4 5 }

A-Probable-Cause	{ 2 9 3 2 7 1 8 }
A-Procedural-Status	{ 2 9 3 2 7 3 6 }
A-Proposed-Repair-Actions	{ 2 9 3 2 7 1 9 }
A-Provider-Object	{ 2 9 3 2 7 4 6 }
A-Relationship-Change-Definition	{ 2 9 3 2 7 2 0 }
A-Scheduler-Name	{ 2 9 3 2 7 6 7 }
A-Secondary	{ 2 9 3 2 7 4 7 }
A-Security-Alarm-Cause	{ 2 9 3 2 7 2 1 }
A-Security-Alarm-Detector	{ 2 9 3 2 7 2 2 }
A-Security-Alarm-Severity	{ 2 9 3 2 7 2 3 }
A-Service-Provider	{ 2 9 3 2 7 2 4 }
A-Service-User	{ 2 9 3 2 7 2 5 }
A-Source-Indicator	{ 2 9 3 2 7 2 6 }
A-Specific-Problems	{ 2 9 3 2 7 2 7 }
A-Standby-Status	{ 2 9 3 2 7 3 7 }
A-Start-Time	{ 2 9 3 2 7 6 8 }
A-State-Change-Definition	{ 2 9 3 2 7 2 8 }
A-Stop-Time	{ 2 9 3 2 7 6 9 }
A-Supported-Features	{ 2 9 3 2 7 7 0 }
A-System-Id	{ 2 9 3 2 7 4 }
A-System-Title	{ 2 9 3 2 7 5 }
A-Threshold-Info	{ 2 9 3 2 7 2 9 }
A-Trend-Indication	{ 2 9 3 2 7 3 0 }
A-Unknown-Status	{ 2 9 3 2 7 3 8 }
A-Usage-State	{ 2 9 3 2 7 3 9 }
A-User-Object	{ 2 9 3 2 7 4 8 }
A-Week-Mask	{ 2 9 3 2 7 7 1 }

A.2.4 Object Identifiers for DMI Attribute Groups

Attribute Group	Object Identifier
A-Relationships	{ 2 9 3 2 8 2 }
A-State	{ 2 9 3 2 8 1 }

A.2.5 Object Identifiers for DMI Notifications

Notification	Object Identifier
N-Attribute-Value-Change	{ 2 9 3 2 1 0 1 }
N-Communications-Alarm	{ 2 9 3 2 1 0 2 }
N-Environmental-Alarm	{ 2 9 3 2 1 0 3 }
N-Equipment-Alarm	{ 2 9 3 2 1 0 4 }
N-Integrity-Violation	{ 2 9 3 2 1 0 5 }
N-Object-Creation	{ 2 9 3 2 1 0 6 }
N-Object-Deletion	{ 2 9 3 2 1 0 7 }
N-Operational-Violation	{ 2 9 3 2 1 0 8 }

N-Physical-Violation	{ 2 9 3 2 10 9 }
N-Processing-Error-Alarm	{ 2 9 3 2 10 10 }
N-Qualityof-Service-Alarm	{ 2 9 3 2 10 11 }
N-Relationship-Change	{ 2 9 3 2 10 12 }
N-Security-Service-Or-Mechanism-Violation	{ 2 9 3 2 10 13 }
N-State-Change	{ 2 9 3 2 10 14 }
N-Time-Domain-Violation	{ 2 9 3 2 10 15 }

A.2.6 Object Identifiers for DMI Parameters

Parameter	Object Identifier
S-Miscellaneous-Error	{ 2 9 3 2 5 1 }

A.3 DMI Attribute Value Syntaxes

The OM attribute *attribute-Value* defined by XMP may have one of the following forms, depending on its OM attribute *attribute-Id*.

Attribute Type	Attribute Syntax
A-Active-Destination	Object(Destination)
A-Additional-Information	Object(Additional-Information)
A-Additional-Text	String(Graphic)
A-Administrative-State	Enum(Administrative-State)
A-Alarm-Status	Object(Alarm-Status)
A-Allomorphs	Object(Allomorphs)
A-Attribute-Identifier-List	Object(Attribute-Identifier-List)
A-Attribute-List	Object(Attribute-List)
A-Attribute-Value-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)
A-Availability-Status	Object(Availability-Status)
A-Backed-Up-Object	Object(Back-Up-Relationship-Object)
A-Backed-Up-Status	Boolean
A-Back-Up-Destination-List	Object(Back-Up-Destination-List)
A-Back-Up-Object	Object(Back-Up-Relationship-Object)
A-Capacity-Alarm-Threshold	Object(Capacity-Alarm-Threshold)
A-Confirmed-Mode	Boolean
A-Control-Status	Object(Control-Status)
A-Correlated-Notifications	Object(Setof-Correlated-Notifications)
A-Corrupted-PD-Us-Received-Counter	Integer
A-Corrupted-PD-Us-Received-Threshold	Object(Setof-Counter-Threshold)
A-Counter	Integer
A-Counter-Threshold	Object(Setof-Counter-Threshold)
A-Current-Log-Size	Integer
A-Destination	Object(Destination)
A-Discriminator-Construct	Object(CMIS-Filter)
A-Discriminator-Id	Object(Simple-Name-Type)
A-Event-Time	String(Generalised-Time)
A-Event-Type	Object(Event-Type-Id)
A-Gauge	Object(Observed-Value)
A-Gauge-Threshold	Object(Setof-Gauge-Threshold)
A-Incoming-Connection-Reject-Error-Counter	Integer
A-Incoming-Connection-Reject-Error-Threshold	Object(Setof-Counter-Threshold)
A-Incoming-Connection-Requests-Counter	Integer
A-Incoming-Connection-Requests-Threshold	Object(Setof-Counter-Threshold)
A-Incoming-Disconnect-Counter	Integer
A-Incoming-Disconnect-Error-Counter	Integer
A-Incoming-Disconnect-Error-Threshold	Object(Setof-Counter-Threshold)
A-Incoming-Protocol-Error-Counter	Integer
A-Incoming-Protocol-Error-Threshold	Object(Setof-Counter-Threshold)
A-Intervals-Of-Day	Object(Setof-Intervals-Of-Day)
A-Log-Full-Action	Enum(Log-Full-Action)
A-Logging-Time	String(Generalised-Time)
A-Log-Id	Object(Simple-Name-Type)
A-Log-Record-Id	Object(Simple-Name-Type)
A-Managed-Object-Class	Object(Object-Class)
A-Managed-Object-Instance	Object(Object-Instance)
A-Max-Log-Size	Integer

A-Member	Object(Group-Objects)
A-Monitored-Attributes	Object(Monitored-Attributes)
A-Name-Binding	String(Object-Identifier)
A-Notification-Identifier	Integer
A-Number-Of-Records	Integer
A-Object-Class	Object(Object-Class)
A-Octets-Received-Counter	Integer
A-Octets-Received-Threshold	Object(Setof-Counter-Threshold)
A-Octets-Retransmitted-Error-Counter	Integer
A-Octets-Retransmitted-Threshold	Object(Setof-Counter-Threshold)
A-Octets-Sent-Counter	Integer
A-Octets-Sent-Threshold	Object(Setof-Counter-Threshold)
A-Operational-State	Enum(Operational-State)
A-Outgoing-Connection-Reject-Error-Counter	Integer
A-Outgoing-Connection-Reject-Error-Threshold	Object(Setof-Counter-Threshold)
A-Outgoing-Connection-Requests-Counter	Integer
A-Outgoing-Connection-Requests-Threshold	Object(Setof-Counter-Threshold)
A-Outgoing-Disconnect-Counter	Integer
A-Outgoing-Disconnect-Error-Counter	Integer
A-Outgoing-Disconnect-Error-Threshold	Object(Setof-Counter-Threshold)
A-Outgoing-Protocol-Error-Counter	Integer
A-Outgoing-Protocol-Error-Threshold	Object(Setof-Counter-Threshold)
A-Owner	Object(Group-Objects)
A-Packages	Object(Packages)
A-PD-Us-Received-Counter	Integer
A-PD-Us-Received-Threshold	Object(Setof-Counter-Threshold)
A-PD-Us-Retransmitted-Error-Counter	Integer
A-PD-Us-Retransmitted-Error-Threshold	Object(Setof-Counter-Threshold)
A-PD-Us-Sent-Counter	Integer
A-PD-Us-Sent-Threshold	Object(Setof-Counter-Threshold)
A-Peer	Object(Back-Up-Relationship-Object)
A-Perceived-Severity	Enum(Perceived-Severity)
A-Primary	Object(Setof-Prioritised-Object)
A-Probable-Cause	Object(Probable-Cause)
A-Procedural-Status	Object(Procedural-Status)
A-Proposed-Repair-Actions	Object(Proposed-Repair-Actions)
A-Provider-Object	Object(Setof-Prioritised-Object)
A-Relationship-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)
A-Scheduler-Name	Object(Object-Instance)
A-Secondary	Object(Setof-Prioritised-Object)
A-Security-Alarm-Cause	String(Object-Identifier)
A-Security-Alarm-Detector	Object(Security-Alarm-Detector)
A-Security-Alarm-Severity	Enum(Perceived-Severity)
A-Service-Provider	Object(Service-User)
A-Service-User	Object(Service-User)
A-Source-Indicator	Enum(Source-Indicator)
A-Specific-Problems	Object(Specific-Problems)
A-Standby-Status	Integer
A-Start-Time	String(Generalised-Time)
A-State-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)
A-Stop-Time	Object(Stop-Time)
A-Supported-Features	Object(Setof-Supported-Features)
A-System-Id	Object(System-Id)
A-System-Title	Object(System-Title)
A-Threshold-Info	Object(Threshold-Info)

A-Tidemark	Object(Tide-Mark-Info)
A-Trend-Indication	Enum(Trend-Indication)
A-Unknown-Status	Boolean
A-Usage-State	Enum(Usage-State)
A-User-Object	Object(Setof-Prioritised-Object)
A-Week-Mask	Object(Setof-Week-Mask)

A.4 DMI Notification Information Syntaxes

The *Event-Info* OM attribute of the *CMIS-Event-Report-Argument* may have one of the following form, depending on its OM attribute *Event-Type*:

Notification Type	Information Syntax	Reply Syntax
N-Attribute-Value-Change	Object(Attribute-Value-Change-Info)	-
N-Communications-Alarm	Object(Alarm-Info)	-
N-Environmental-Alarm	Object(Alarm-Info)	-
N-Equipment-Alarm	Object(Alarm-Info)	-
N-Integrity-Violation	Object(Security-Alarm-Info)	-
N-Object-Creation	Object(Object-Info)	-
N-Object-Deletion	Object(Object-Info)	-
N-Operational-Violation	Object(Security-Alarm-Info)	-
N-Physical-Violation	Object(Security-Alarm-Info)	-
N-Processing-Error-Alarm	Object(Alarm-Info)	-
N-Qualityof-Service-Alarm	Object(Alarm-Info)	-
N-Relationship-Change	Object(Relationship-Change-Info)	-
N-Security-Service-Or-Mechanism-Violation	Object(Security-Alarm-Info)	-
N-State-Change	Object(State-Change-Info)	-
N-Time-Domain-Violation	Object(Security-Alarm-Info)	-

A.5 DMI Parameter Syntaxes

The following Parameter Syntaxes are defined, according to the DMI specifications.

Parameter Type	Parameter Syntax
S-Miscellaneous-Error	Null

A.6 OM Class Hierarchy and Attribute Tables

This section defines the additional OM classes used to represent values of the DMI attributes, notifications, and parameters described in the Interface Class Definitions chapter of the Management Protocols API (reference **XMP**).

The list below depicts the hierarchical organisation of the OM classes which are defined in the following sections, and thus shows which OM classes inherit additional OM attributes from their OM superclasses. Subclassification is indicated by indentation, and the names of abstract OM classes are rendered in italics. Thus, for example, **Alarm-Info** is an immediate subclass of the abstract OM class *Object*.

Object

- **Additional-Information**
- **Alarm-Info**
- **Alarm-Status**
- **Allomorphs**
- **Attribute-Identifier-List**
- **Attribute-List**
- **Attribute-Value-Change-Definition**
- **Attribute-Value-Change-Info**
- **Availability-Status**
- **Back-Up-Destination-List**
- **Back-Up-Relationship-Object**
- **Capacity-Alarm-Threshold**
- **Control-Status**
- **Correlated-Notifications**
- **Correlated-Notifications-1**
- **Counter-Threshold**
- **Destination**
- **Down**
- **Gauge-Threshold**
- **Group-Objects**
- **Intervals-Of-Day**
- **Management-Extension**
- **Monitored-Attributes**
- **Multiple**
- **Notify-Threshold**
- **Object-Info**
- **Observed-Value**
- **Packages**
- **Prioritised-Object**
- **Probable-Cause**
- **Procedural-Status**
- **Proposed-Repair-Actions**
- **Relationship-Change-Info**
- **Security-Alarm-Detector**
- **Security-Alarm-Info**

- **Service-User**
- **Setof-Attribute-Value-Change-Definition**
- **Setof-Correlated-Notifications**
- **Setof-Counter-Threshold**
- **Setof-Gauge-Threshold**
- **Setof-Intervals-of-Day**
- **Setof-Prioritised-Object**
- **Setof-Supported-Features**
- **Setof-Week-Mask**
- **Simple-Name-Type**
- **Specific-Identifier**
- **Specific-Problems**
- **State-Change-Info**
- **Stop-Time**
- **Supported-Features**
- **System-Id**
- **System-Title**
- **Threshold-Info**
- **Threshold-Level-Ind**
- **Tide-Mark**
- **Tide-Mark-Info**
- **Time24**
- **Up**
- **Week-Mask.**

The Attribute Tables for these OM classes are presented on the following pages.

Additional-Information

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
management-Extension	Object(Management-Extension)	-	0-more

Table A-1 OM Attributes of Additional-Information**Alarm-Info**

An instance of OM class Alarm-Info contains the information commonly supplied with an event of any type of Alarm. The semantic of the OM attributes pertaining to the Alarm-Info OM class is given in ISO/IEC 10164-4 (reference AM).

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
probable-Cause	Object(Probable-Cause)	-	1
specific-Problems	Object(Specific-Problems)	-	0-1
perceived-Severity	Enum(Perceived-Severity)	-	1
backed-Up-Status	Boolean	-	0-1
back-Up-Object	Object(Object-Instance)	-	0-1
trend-Indication	Enum(Trend-Indication)	-	0-1
threshold-Info	Object(Threshold-Info)	-	0-1
notification-Identifier	Integer	-	0-1
correlated-Notifications	Object(Setof-Correlated-Notifications)	-	0-1
state-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)	-	0-1
monitored-Attributes	Object(Monitored-Attributes)	-	0-1
proposed-Repair-Actions	Object(Proposed-Repair-Actions)	-	0-1
additional-Text	String(Graphic)	-	0-1
additional-Information	Object(Additional-Information)	-	0-1

Table A-2 OM Attributes of Alarm-Info**Alarm-Status**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
alarm-Status	Integer	-	0-more

Table A-3 OM Attributes of Alarm-Status

Allomorphs

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
object-Class	Object(Object-Class)	-	0-more

Table A-4 OM Attributes of Allomorphs**Attribute-Identifier-List**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-Id	Object(Attribute-Id)	-	0-more

Table A-5 OM Attributes of Attribute-Identifier-List**Attribute-List**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute	Object(Attribute)	-	0-more

Table A-6 OM Attributes of Attribute-List**Attribute-Value-Change-Definition**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-ID	Object(Attribute-Id)	-	1
old-Attribute-Value	any	-	0-1
new-Attribute-Value	any	-	1

Table A-7 OM Attributes of Attribute-Value-Change-Definition

Attribute-Value-Change-Info

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
source-Indicator	Enum(Source-Indicator)	-	0-1
attribute-Identifier-List	Object(Attribute-Identifier-List)	-	0-1
attribute-Value-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)	-	1
notification-Identifier	Integer	-	0-1
correlated-Notifications	Object(Setof-Correlated-Notifications)	-	0-1
additional-Text	String(Graphic)	-	0-1
additional-Information	Object(Additional-Information)	-	0-1

Table A-8 OM Attributes of Attribute-Change-Info**Availability-Status**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
availability-Status	Integer	-	0-more

Table A-9 OM Attributes of Availability-Status**Back-Up-Destination-List**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
a-E-Title	Object(AE-Title)	-	0-more

Table A-10 OM Attributes of Back-Up-Destination-List**Back-Up-Relationship-Object**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
object-Name	Object(Object-Instance)	-	0-1
no-Object	Null	-	0-1

Table A-11 OM Attributes of Back-Up-Relationship-Object

Capacity-Alarm-Threshold

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
capacity-Alarm-Threshold	Integer	-	0-more

Table A-12 OM Attributes of Capacity-Alarm-Threshold**Control-Status**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
control-Status	Integer	-	0-more

Table A-13 OM Attributes of Control-Status**Correlated-Notifications**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
correlated-Notifications	Object(Correlated-Notifications-1)	-	1
source-Object-Inst	Object(Object-Instance)	-	0-1

Table A-14 OM Attributes of Correlated-Notifications**Correlated-Notifications-1**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
notification-Identifier	Integer	-	0-more

Table A-15 OM Attributes of Correlated-Notifications-1

Counter-Threshold

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
comparison-Level	Integer	-	1
offset-Value	Integer	-	1
notification-On-Off	Boolean	-	1

Table A-16 OM Attributes of Counter-Threshold

Destination

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
single	Object(AE-Title)	-	0-1
multiple	Object(Multiple)	-	0-1

Table A-17 OM Attributes of Destination

Down

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
high	Object(Observed-Value)	-	1
low	Object(Observed-Value)	-	1

Table A-18 OM Attributes of Down

Gauge-Threshold

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
notify-Low	Object(Notify-Threshold)	-	1
notify-High	Object(Notify-Threshold)	-	1

Table A-19 OM Attributes of Gauge-Threshold

Group-Objects

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
object-Instance	Object(Object-Instance)	-	0-more

Table A-20 OM Attributes of Group-Objects**Intervals-of-Day**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
interval-Start	Object(Time24)	-	1
interval-End	Object(Time24)	-	1

Table A-21 OM Attributes of Intervals-of-Day**Management-Extension**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
identifier	String(Object-Identi fier)	-	1
significance	Boolean	-	0-1
information	any	-	1

Table A-22 OM Attributes of Management-Extension**Monitored-Attributes**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute	Object(Attribute)	-	0-more

Table A-23 OM Attributes of Monitored-Attributes

Multiple

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
ae-Title	Object(AE-Title)	-	0-more

Table A-24 OM Attributes of AE-Title**Notify-Threshold**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
threshold	Object(Observed-Value)	-	1
notify-On-Off	Boolean	-	1

Table A-25 OM Attributes of Notify-Threshold**Object-Info**

An instance of OM class Object-Info contains the information supplied with an event of type Object-Creation or Object-Deletion.

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
source-Indicator	Enum(Source-Indicator)	-	0-1
attribute-List	Object(Attribute-List)	-	0-1
notification-Identifier	Integer	-	0-1
correlated-Notifications	Object(Setof-Correlated-Notifications)	-	0-1
additional-Text	String(Graphic)	-	0-1
additional-Information	Object(Additional-Information)	-	0-1

Table A-26 OM Attributes of Object-Info**Observed-Value**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
integer	Integer	-	0-1
real	Real	-	0-1

Table A-27 OM Attributes of Observed-Value

Packages

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
packages	String(Object-Identifier)	-	0-more

Table A-28 OM Attributes of Packages

Prioritised-Object

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
object	Object(Object-Instance)	-	1
priority	Integer	-	1

Table A-29 OM Attributes of Prioritised-Object

Probable-Cause

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
global-Value	String(Object-Identifier)	-	0-1
local-Value	Integer	-	0-1

Table A-30 OM Attributes of Probable-Cause

Procedural-Status

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
procedural-Status	Integer	-	0-more

Table A-31 OM Attributes of Procedural-Status

Proposed-Repair-Actions

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
specific-Identifier	Object(Specific-Identifier)	-	0-more

Table A-32 OM Attributes of Proposed-Repair-Actions

Relationship-Change-Info

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
source-Indicator	Enum(Source-Indicator)	-	0-1
attribute-Identifier-List	Object(Attribute-Identifier-List)	-	0-1
relationship-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)	-	1
notification-Identifier	Integer	-	0-1
correlated-Notifications	Object(Setof-Correlated-Notifications)	-	0-1
additional-Text	String(Graphic)	-	0-1
additional-Information	Object(Additional-Information)	-	0-1

Table A-33 OM Attributes of Relationship-Change-Info

Security-Alarm-Detector

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
mechanism	String(Object-Identifier)	-	0-1
object	Object(Object-Instance)	-	0-1
application	Object(AE-Title)	-	0-1

Table A-34 OM Attributes of Security-Alarm-Detector

Security-Alarm-Info

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
security-Alarm-Cause	String(Object-Identi fier)	-	1
security-Alarm-Severity	Enum(Perceived-Severity)	-	1
security-Alarm-Detector	Object(Security-Alarm-Detector)	-	1
service-User	Object(Service-User)	-	1
service-Provider	Object(Service-User)	-	1
notification-Identi fier	Integer	-	0-1
correlated-Noti fications	Object(Setof-Correlated-Noti fications)	-	0-1
additional-Text	String(Graphic)	-	0-1
additional-Information	Object(Additional-Information)	-	0-1

Table A-35 OM Attributes of Security-Alarm-Info

Service-User

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
identifier	String(Object-Identi fier)	-	1
details	any	-	1

Table A-36 OM Attributes of Service-User

Setof-Attribute-Value-Change-Definition

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
attribute-Value-Change-De finition	Object(Attribute-Value-Change-De finition)	-	0-more

Table A-37 OM Attributes of Setof-Attribute-Value-Change-Definition

Setof-Correlated-Notifications

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
correlated-Noti fications	Object(Correlated-Noti fications)	-	0-more

Table A-38 OM Attributes of Setof-Correlated-Notifications

Setof-Counter-Threshold

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
counter-Threshold	Object(Counter-Threshold)	-	0-more

Table A-39 OM Attributes of Setof-Counter-Threshold**Setof-Gauge-Threshold**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
gauge-Threshold	Object(Gauge-Threshold)	-	0-more

Table A-40 OM Attributes of Setof-Gauge-Threshold**Setof-Intervals-Of-Day**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
intervals-Of-Day	Object(Intervals-Of-Day)	-	0-more

Table A-41 OM Attributes of Setof-Intervals-Of-Day**Setof-Prioritised-Object**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
prioritised-Object	Object(Prioritised-Object)	-	0-more

Table A-42 OM Attributes of Setof-Prioritised-Object

Setof-Supported-Features

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
supported-features	Object(Supported-Features)	-	0-more

Table A-43 OM Attributes of Setof-Supported-Features**Setof-Week-Mask**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
week-Mask	Object(Week-Mask)	-	0-more

Table A-44 OM Attributes of Setof-Week-Mask**Simple-Name-Type**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
number	Integer	-	0-1
string	String(Graphic)	-	0-1

Table A-45 OM Attributes of Simple-Name-Type**Specific-Identifier**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
specific-Identifier-OBJECT-IDENTIFIER-1	String(Object-Identifier)	-	0-1
specific-Identifier-INTEGER-2	Integer	-	0-1

Table A-46 OM Attributes of Specific-Identifier

Specific-Problems

An instance of OM class Specific-Problems identifies the problem of a particular alarm.

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
specific-Identifier	Object(Specific-Identifier)	-	0-more

Table A-47 OM Attributes of Specific-Problems

State-Change-Info

An instance of OM class State-Change-Info contains the information supplied with an event of type State-Change.

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
source-Indicator	Enum(Source-Indicator)	-	0-1
attribute-Identifier-List	Object(Attribute-Identifier-List)	-	0-1
state-Change-Definition	Object(Setof-Attribute-Value-Change-Definition)	-	1
notification-Identifier	Integer	-	0-1
correlated-Notifications	Object(Setof-Correlated-Notifications)	-	0-1
additional-Text	String(Graphic)	-	0-1
additional-Information	Object(Additional-Information)	-	0-1

Table A-48 OM Attributes of State-Change-Info

Stop-Time

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
specific	String(Generalised-Time)	-	0-1
continual	Null	-	0-1

Table A-49 OM Attributes of Stop-Time

Supported-Features

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
feature-Identifier	String(Object-Identi fier)	-	1
feature-Info	any	-	1

Table A-50 OM Attributes of Supported-Feature

System-Id

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
name	String(Graphic)	-	0-1
number	Integer	-	0-1
nothing	Null	-	0-1

Table A-51 OM Attributes of System-Id**System-Title**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
distinguished-Name	Object(DS-DN)	-	0-1
oid	String(Object-Identifier)	-	0-1
nothing	Null	-	0-1

Table A-52 OM Attributes of System-Title**Threshold-Info**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
OM Attribute	Value Syntax	Value Length	Value Number
triggered-Threshold	Object(Attribute-Id)	-	1
observed-Value	Object(Observed-Value)	-	1
threshold-Level	Object(Threshold-Level-Ind)	-	0-1
arm-Time	String(Generalised-Time)	-	0-1

Table A-53 OM Attributes of Threshold-Info

Threshold-Level-Ind

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
up	Object(Up)	-	0-1
down	Object(Down)	-	0-1

Table A-54 OM Attributes of Threshold-Level-Ind**Tide-Mark**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below. Exactly one OM attribute is permitted in an instance of this OM class.

OM Attribute	Value Syntax	Value Length	Value Number
max-Tide-Mark	Object(Observed-Value)	-	0-1
min-Tide-Mark	Object(Observed-Value)	-	0-1

Table A-55 OM Attributes of Tide-Mark**Tide-Mark-Info**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
current-Tide-Mark	Object(Tide-Mark)	-	1
previous-Tide-Mark	Object(Tide-Mark)	-	1
reset-Time	String(Generalised-Time)	-	1

Table A-56 OM Attributes of Tide-Mark-Info**Time24**

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
hour	Integer	-	1
minute	Integer	-	1

Table A-57 OM Attributes of Time24

Up

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
high	Object(Observed-Value)	-	1
low	Object(Observed-Value)	-	0-1

Table A-58 OM Attributes of Up

Week-Mask

An instance of this OM class has the OM attributes of its superclass - *Object* - and additionally the OM attributes listed below.

OM Attribute	Value Syntax	Value Length	Value Number
days-Of-Week	String(Bit)	7	1
intervals-Of-Day	Object(Setof-Intervals-Of-Day)	-	1

Table A-59 OM Attributes of Week-Mask

A.7 Attribute Enumeration Lists

Value List for Administrative-State

administrative-State. Its value is one of:

- locked
- unlocked
- shutting-Down.

Value List for Alarm-Status

alarm-Status. Its value is one of:

- under-Repair
- critical
- major
- minor
- alarm outstanding.

Value List for Availability-Status

availability-Status. Its value is one of:

- in test
- failed
- power off
- off line
- off duty
- dependency
- degraded
- not installed
- log full.

Value List for Control-Status

control-Status. Its value is one of:

- subject to test
- part of services locked
- reserved for test
- suspended.

Value List for Days-Of-Week

days-Of-Week. Its value is one of:

- sunday
- monday
- tuesday
- wednesday
- thursday
- friday
- saturday.

Value List for Log-Full-Action

log-Full-Action. Its value is one of:

- wrap
- halt.

Value List for Max-Log-Size

max-Log-Size. Its value is:

- unlimited.

Value List for Operational-State

operational-State. Its value is one of:

- disabled
- enabled.

Value List for Perceived-Severity

perceived-Severity. Its value is one of:

- indeterminate
- critical
- major
- minor
- warning
- cleared.

Value List for Priority

priority. Its value is one of:

- lowest
- highest.

Value List for Probable-Cause

probable-Cause. Its value is one of:

- adapter-Error
- application-Subsystem-Failure
- bandwidth-Reduced
- call-Establishment-Error
- communications-Protocol-Error
- communications-Subsystem-Failure
- configuration-Or-Customization-Error
- congestion
- corrupt-Data
- cPU-Cycles-Limit-Exceeded
- data-Set-Or-Modem-Error
- degraded-Signal
- dTE-DCE-Interface-Error
- enclosure-Door-Open
- equipment-Malfunction
- excessive-Vibration
- file-Error
- fire-Detected
- flood-Detected
- framing-Error
- heating-Or-Ventilation-Or-Cooling-System-Problem
- humidity-Unacceptable
- input-Output-Device-Error
- input-Device-Error
- LAN-Error
- leak-Detected
- local-Node-Transmission-Error
- loss-Of-Frame

- loss-Of-Signal
- material-Supply-Exhausted
- multiplexer-Problem
- out-Of-Memory
- output-Device-Error
- performance-Degraded
- power-Problem
- pressure-Unacceptable
- processor-Problem
- pump-Failure
- queue-Size-Exceeded
- receive-Failure
- receiver-Failure
- remote-Node-Transmission-Error
- resource-At-Or-Nearing-Capacity
- response-Time-Excessive
- retransmission-Rate-Excessive
- software-Error
- software-Program-Abnormally-Terminated
- software-Program-Error
- storage-Capacity-Problem
- temperature-Unacceptable
- threshold-Crossed
- timing-Problem
- toxic-Leak-Detected
- transmit-Failure
- transmitter-Failure
- underlying-Resource-Unavailable
- version-Mismatch.

Value List for Procedural-Status

procedural-Status. Its value is one of:

- initialization required
- not initialized
- initializing
- reporting
- terminating.

Value List for Security-Alarm-Cause

security-Alarm-Cause. Its value is one of:

- authentication-Failure
- breach-Of-Confidentiality
- cable-Tamper
- delayed-Information
- denial-Of-Service
- duplicate-Information
- information-Missing
- information-Modification-Detected
- information-Out-Of-Sequence
- intrusion-Detection
- key-Expired
- non-Repudiation-Failure
- out-Of-Hours-Activity
- out-Of-Service
- procedural-Error
- unauthorized-Access-Attempted
- unexpected-Information
- unspecified-Reason.

Value List for Source-Indicator

source-Indicator. Its value is one of:

- resource-Operation
- management-Operation
- unknown.

Value List for Standby-Status

standby-Status. Its value is one of:

- hot standby
- cold standby
- providing service.

Value List for Trend-Indication

trend-Indication. Its value is one of:

- less-Severe
- no-Change
- more-Severe.

Value List for Usage-State

usage-State. Its value is one of:

- idle
- active
- busy.

A.8 C Header File

This section sets out the symbols which are defined in the `<omp_dmi.h>` header.

Where the values of the symbols are indicated, the values are an integral part of the interface. Where a value is not given, the value on a particular system will be determined by the vendor or by an administrator.

The `<omp_dmi.h>` header declares the interface functions, the structures passed to and from those functions, and the defined constants used by the functions and structures.

All application programs which include this header must first include the OSI-Abstract-Data Manipulation header `<xom.h>`.

All Object Identifiers are represented by constants defined in the headers. These constants are used with the macros defined in the XOM API (reference **XOM**). A constant is defined to represent the Object Identifier of the DMI Management Contents package.

Every application program which makes use of a class or other Object Identifier must explicitly import it into every compilation unit (C source program) which uses it. Each such class or Object Identifier name must be explicitly exported from just one compilation unit.

In the header file, OM class constants are prefixed with `OMP_O_` to denote that they are OM classes. However, when using the `OM_IMPORT` and `OM_EXPORT` macros, the base names (without the `OMP_O_` prefix) should be used. For example:

```
OM_IMPORT(C_DMI_ALARM_STATUS)
```

```
/* DMI Package Definition */
#define OMP_O_SMI_PART2  "\x2A\x86\x3A\x00\x88\x1A\x06\x04"

/* DMI Managed Object OID Definitions */
#define OMP_O_DMI_O_ALARM_RECORD  "\x59\x03\x02\x03\x01"
#define OMP_O_DMI_O_ATTRIBUTE_VALUE_CHANGE_RECORD  "\x59\x03\x02\x03\x02"
#define OMP_O_DMI_O_DISCRIMINATOR  "\x59\x03\x02\x03\x03"
#define OMP_O_DMI_O_EVENT_FORWARDING_DISCRIMINATOR  "\x59\x03\x02\x03\x04"
#define OMP_O_DMI_O_EVENT_LOG_RECORD  "\x59\x03\x02\x03\x05"
#define OMP_O_DMI_O_LOG  "\x59\x03\x02\x03\x06"
#define OMP_O_DMI_O_LOG_RECORD  "\x59\x03\x02\x03\x07"
#define OMP_O_DMI_O_OBJECT_CREATION_RECORD  "\x59\x03\x02\x03\x08"
#define OMP_O_DMI_O_OBJECT_DELETION_RECORD  "\x59\x03\x02\x03\x09"
#define OMP_O_DMI_O_RELATIONSHIP_CHANGE_RECORD  "\x59\x03\x02\x03\x0A"
#define OMP_O_DMI_O_SECURITY_ALARM_REPORT_RECORD  "\x59\x03\x02\x03\x0B"
#define OMP_O_DMI_O_STATE_CHANGE_RECORD  "\x59\x03\x02\x03\x0C"
#define OMP_O_DMI_O_SYSTEM  "\x59\x03\x02\x03\x0D"
#define OMP_O_DMI_O_TOP  "\x59\x03\x02\x03\x0E"

/* DMI Attribute OID Definitions */
#define OMP_O_DMI_A_ACTIVE_DESTINATION  "\x59\x03\x02\x07\x31"
#define OMP_O_DMI_A_ADDITIONAL_INFORMATION  "\x59\x03\x02\x07\x06"
#define OMP_O_DMI_A_ADDITIONAL_TEXT  "\x59\x03\x02\x07\x07"
#define OMP_O_DMI_A_ADMINISTRATIVE_STATE  "\x59\x03\x02\x07\x1F"
#define OMP_O_DMI_A_ALARM_STATUS  "\x59\x03\x02\x07\x20"
#define OMP_O_DMI_A_ALLOMORPHS  "\x59\x03\x02\x07\x32"
#define OMP_O_DMI_A_ATTRIBUTE_IDENTIFIER_LIST  "\x59\x03\x02\x07\x08"
#define OMP_O_DMI_A_ATTRIBUTE_LIST  "\x59\x03\x02\x07\x09"
```

```

#define OMP_O_DMI_A_ATTRIBUTE_VALUE_CHANGE_DEFINITION "\x59\x03\x02\x07\x0A"
#define OMP_O_DMI_A_AVAILABILITY_STATUS "\x59\x03\x02\x07\x21"
#define OMP_O_DMI_A_BACK_UP_DESTINATION_LIST "\x59\x03\x02\x07\x33"
#define OMP_O_DMI_A_BACK_UP_OBJECT "\x59\x03\x02\x07\x28"
#define OMP_O_DMI_A_BACKED_UP_OBJECT "\x59\x03\x02\x07\x29"
#define OMP_O_DMI_A_BACKED_UP_STATUS "\x59\x03\x02\x07\x0B"
#define OMP_O_DMI_A_CAPACITY_ALARM_THRESHOLD "\x59\x03\x02\x07\x34"
#define OMP_O_DMI_A_CONFIRMED_MODE "\x59\x03\x02\x07\x35"
#define OMP_O_DMI_A_CONTROL_STATUS "\x59\x03\x02\x07\x22"
#define OMP_O_DMI_A_CORRELATED_NOTIFICATIONS "\x59\x03\x02\x07\x0C"
#define OMP_O_DMI_A_CORRUPTED_PD_US_RECEIVED_COUNTER "\x59\x03\x02\x07\x48"
#define OMP_O_DMI_A_CORRUPTED_PD_US_RECEIVED_THRESHOLD "\x59\x03\x02\x07\x59"
#define OMP_O_DMI_A_CURRENT_LOG_SIZE "\x59\x03\x02\x07\x36"
#define OMP_O_DMI_A_DESTINATION "\x59\x03\x02\x07\x37"
#define OMP_O_DMI_A_DISCRIMINATOR_CONSTRUCT "\x59\x03\x02\x07\x38"
#define OMP_O_DMI_A_DISCRIMINATOR_ID "\x59\x03\x02\x07\x01"
#define OMP_O_DMI_A_EVENT_TIME "\x59\x03\x02\x07\x0D"
#define OMP_O_DMI_A_EVENT_TYPE "\x59\x03\x02\x07\x0E"
#define OMP_O_DMI_A_INCOMING_CONNECTION_REJECT_ERROR_COUNTER "\x59\x03\x02\x07\x49"
#define OMP_O_DMI_A_INCOMING_CONNECTION_REJECT_ERROR_THRESHOLD "\x59\x03\x02\x07\x5A"
#define OMP_O_DMI_A_INCOMING_CONNECTION_REQUESTS_COUNTER "\x59\x03\x02\x07\x4A"
#define OMP_O_DMI_A_INCOMING_CONNECTION_REQUESTS_THRESHOLD "\x59\x03\x02\x07\x5B"
#define OMP_O_DMI_A_INCOMING_DISCONNECT_COUNTER "\x59\x03\x02\x07\x4B"
#define OMP_O_DMI_A_INCOMING_DISCONNECT_ERROR_COUNTER "\x59\x03\x02\x07\x4C"
#define OMP_O_DMI_A_INCOMING_DISCONNECT_ERROR_THRESHOLD "\x59\x03\x02\x07\x5C"
#define OMP_O_DMI_A_INCOMING_PROTOCOL_ERROR_COUNTER "\x59\x03\x02\x07\x4D"
#define OMP_O_DMI_A_INCOMING_PROTOCOL_ERROR_THRESHOLD "\x59\x03\x02\x07\x5D"
#define OMP_O_DMI_A_INTERVALS_OF_DAY "\x59\x03\x02\x07\x39"
#define OMP_O_DMI_A_LOG_FULL_ACTION "\x59\x03\x02\x07\x3A"
#define OMP_O_DMI_A_LOG_ID "\x59\x03\x02\x07\x02"
#define OMP_O_DMI_A_LOG_RECORD_ID "\x59\x03\x02\x07\x03"
#define OMP_O_DMI_A_LOGGING_TIME "\x59\x03\x02\x07\x3B"
#define OMP_O_DMI_A_MANAGED_OBJECT_CLASS "\x59\x03\x02\x07\x3C"
#define OMP_O_DMI_A_MANAGED_OBJECT_INSTANCE "\x59\x03\x02\x07\x3D"
#define OMP_O_DMI_A_MAX_LOG_SIZE "\x59\x03\x02\x07\x3E"
#define OMP_O_DMI_A_MEMBER "\x59\x03\x02\x07\x2A"
#define OMP_O_DMI_A_MONITORED_ATTRIBUTES "\x59\x03\x02\x07\x0F"
#define OMP_O_DMI_A_NAME_BINDING "\x59\x03\x02\x07\x3F"
#define OMP_O_DMI_A_NOTIFICATION_IDENTIFIER "\x59\x03\x02\x07\x10"
#define OMP_O_DMI_A_NUMBER_OF_RECORDS "\x59\x03\x02\x07\x40"
#define OMP_O_DMI_A_OBJECT_CLASS "\x59\x03\x02\x07\x41"
#define OMP_O_DMI_A_OCTETS_RECEIVED_COUNTER "\x59\x03\x02\x07\x4E"
#define OMP_O_DMI_A_OCTETS_RECEIVED_THRESHOLD "\x59\x03\x02\x07\x5E"
#define OMP_O_DMI_A_OCTETS_RETRANSMITTED_ERROR_COUNTER "\x59\x03\x02\x07\x4F"
#define OMP_O_DMI_A_OCTETS_RETRANSMITTED_THRESHOLD "\x59\x03\x02\x07\x5F"
#define OMP_O_DMI_A_OCTETS_SENT_COUNTER "\x59\x03\x02\x07\x50"
#define OMP_O_DMI_A_OCTETS_SENT_THRESHOLD "\x59\x03\x02\x07\x60"
#define OMP_O_DMI_A_OPERATIONAL_STATE "\x59\x03\x02\x07\x23"
#define OMP_O_DMI_A_OUTGOING_CONNECTION_REJECT_ERROR_COUNTER "\x59\x03\x02\x07\x51"
#define OMP_O_DMI_A_OUTGOING_CONNECTION_REJECT_ERROR_THRESHOLD "\x59\x03\x02\x07\x61"
#define OMP_O_DMI_A_OUTGOING_CONNECTION_REQUESTS_COUNTER "\x59\x03\x02\x07\x52"
#define OMP_O_DMI_A_OUTGOING_CONNECTION_REQUESTS_THRESHOLD "\x59\x03\x02\x07\x62"
#define OMP_O_DMI_A_OUTGOING_DISCONNECT_COUNTER "\x59\x03\x02\x07\x53"
#define OMP_O_DMI_A_OUTGOING_DISCONNECT_ERROR_COUNTER "\x59\x03\x02\x07\x54"
#define OMP_O_DMI_A_OUTGOING_DISCONNECT_ERROR_THRESHOLD "\x59\x03\x02\x07\x63"
#define OMP_O_DMI_A_OUTGOING_PROTOCOL_ERROR_COUNTER "\x59\x03\x02\x07\x55"
#define OMP_O_DMI_A_OUTGOING_PROTOCOL_ERROR_THRESHOLD "\x59\x03\x02\x07\x64"
#define OMP_O_DMI_A_OWNER "\x59\x03\x02\x07\x2B"
#define OMP_O_DMI_A_PACKAGES "\x59\x03\x02\x07\x42"
#define OMP_O_DMI_A_PDUS_RECEIVED_COUNTER "\x59\x03\x02\x07\x56"
#define OMP_O_DMI_A_PDUS_RECEIVED_THRESHOLD "\x59\x03\x02\x07\x65"

```

```

#define OMP_O_DMI_A_PDUS_RETRANSMITTED_ERROR_COUNTER           "\x59\x03\x02\x07\x57"
#define OMP_O_DMI_A_PDUS_RETRANSMITTED_ERROR_THRESHOLD        "\x59\x03\x02\x07\x66"
#define OMP_O_DMI_A_PDUS_SENT_COUNTER                          "\x59\x03\x02\x07\x58"
#define OMP_O_DMI_A_PDUS_SENT_THRESHOLD                       "\x59\x03\x02\x07\x67"
#define OMP_O_DMI_A_PEER                                       "\x59\x03\x02\x07\x2C"
#define OMP_O_DMI_A_PERCEIVED_SEVERITY                        "\x59\x03\x02\x07\x11"
#define OMP_O_DMI_A_PRIMARY                                    "\x59\x03\x02\x07\x2D"
#define OMP_O_DMI_A_PROBABLE_CAUSE                            "\x59\x03\x02\x07\x12"
#define OMP_O_DMI_A_PROCEDURAL_STATUS                         "\x59\x03\x02\x07\x24"
#define OMP_O_DMI_A_PROPOSED_REPAIR_ACTIONS                   "\x59\x03\x02\x07\x13"
#define OMP_O_DMI_A_PROVIDER_OBJECT                           "\x59\x03\x02\x07\x2E"
#define OMP_O_DMI_A_RELATIONSHIP_CHANGE_DEFINITION            "\x59\x03\x02\x07\x14"
#define OMP_O_DMI_A_SCHEDULER_NAME                            "\x59\x03\x02\x07\x43"
#define OMP_O_DMI_A_SECONDARY                                 "\x59\x03\x02\x07\x2F"
#define OMP_O_DMI_A_SECURITY_ALARM_CAUSE                      "\x59\x03\x02\x07\x15"
#define OMP_O_DMI_A_SECURITY_ALARM_DETECTOR                  "\x59\x03\x02\x07\x16"
#define OMP_O_DMI_A_SECURITY_ALARM_SEVERITY                   "\x59\x03\x02\x07\x17"
#define OMP_O_DMI_A_SERVICE_PROVIDER                          "\x59\x03\x02\x07\x18"
#define OMP_O_DMI_A_SERVICE_USER                               "\x59\x03\x02\x07\x19"
#define OMP_O_DMI_A_SOURCE_INDICATOR                          "\x59\x03\x02\x07\x1A"
#define OMP_O_DMI_A_SPECIFIC_PROBLEMS                          "\x59\x03\x02\x07\x1B"
#define OMP_O_DMI_A_STANDBY_STATUS                            "\x59\x03\x02\x07\x25"
#define OMP_O_DMI_A_START_TIME                                 "\x59\x03\x02\x07\x44"
#define OMP_O_DMI_A_STATE_CHANGE_DEFINITION                    "\x59\x03\x02\x07\x1C"
#define OMP_O_DMI_A_STOP_TIME                                  "\x59\x03\x02\x07\x45"
#define OMP_O_DMI_A_SUPPORTED_FEATURES                        "\x59\x03\x02\x07\x46"
#define OMP_O_DMI_A_SYSTEM_ID                                  "\x59\x03\x02\x07\x04"
#define OMP_O_DMI_A_SYSTEM_TITLE                               "\x59\x03\x02\x07\x05"
#define OMP_O_DMI_A_THRESHOLD_INFO                             "\x59\x03\x02\x07\x1D"
#define OMP_O_DMI_A_TREND_INDICATION                           "\x59\x03\x02\x07\x1E"
#define OMP_O_DMI_A_UNKNOWN_STATUS                            "\x59\x03\x02\x07\x26"
#define OMP_O_DMI_A_USAGE_STATE                                "\x59\x03\x02\x07\x27"
#define OMP_O_DMI_A_USER_OBJECT                                "\x59\x03\x02\x07\x30"
#define OMP_O_DMI_A_WEEK_MASK                                  "\x59\x03\x02\x07\x47"

/* DMI Attribute Group OID Definitions */
#define OMP_O_DMI_A_RELATIONSHIPS                             "\x59\x03\x02\x08\x02"
#define OMP_O_DMI_A_STATE                                     "\x59\x03\x02\x08\x01"

/* DMI Notification OID Definitions */
#define OMP_O_DMI_N_ATTRIBUTE_VALUE_CHANGE                    "\x59\x03\x02\x0A\x01"
#define OMP_O_DMI_N_COMMUNICATIONS_ALARM                      "\x59\x03\x02\x0A\x02"
#define OMP_O_DMI_N_ENVIRONMENTAL_ALARM                       "\x59\x03\x02\x0A\x03"
#define OMP_O_DMI_N_EQUIPMENT_ALARM                           "\x59\x03\x02\x0A\x04"
#define OMP_O_DMI_N_INTEGRITY_VIOLATION                       "\x59\x03\x02\x0A\x05"
#define OMP_O_DMI_N_OBJECT_CREATION                           "\x59\x03\x02\x0A\x06"
#define OMP_O_DMI_N_OBJECT_DELETION                           "\x59\x03\x02\x0A\x07"
#define OMP_O_DMI_N_OPERATIONAL_VIOLATION                     "\x59\x03\x02\x0A\x08"
#define OMP_O_DMI_N_PHYSICAL_VIOLATION                       "\x59\x03\x02\x0A\x09"
#define OMP_O_DMI_N_PROCESSING_ERROR_ALARM                     "\x59\x03\x02\x0A\x0A"
#define OMP_O_DMI_N_QUALITYOF_SERVICE_ALARM                    "\x59\x03\x02\x0A\x0B"
#define OMP_O_DMI_N_RELATIONSHIP_CHANGE                       "\x59\x03\x02\x0A\x0C"
#define OMP_O_DMI_N_SECURITY_SERVICE_OR_MECHANISM_VIOLATION   "\x59\x03\x02\x0A\x0D"
#define OMP_O_DMI_N_STATE_CHANGE                               "\x59\x03\x02\x0A\x0E"
#define OMP_O_DMI_N_TIME_DOMAIN_VIOLATION                     "\x59\x03\x02\x0A\x0F"

/* DMI Parameter OID Definitions */
#define OMP_O_DMI_S_MISCELLANEOUS_ERROR                       "\x59\x03\x02\x05\x01"

/* Intermediate object identifier macro */
#define mpP_dmi(X)      (OMP_O_SMI_PART2##X)

```



```

/* OM Class Object Identifiers */
#define OMP_O_C_DMI_ADDITIONAL_INFORMATION mpP_dmi(\xD8\x25)
#define OMP_O_C_DMI_ALARM_INFO mpP_dmi(\xD8\x26)
#define OMP_O_C_DMI_ALARM_STATUS mpP_dmi(\xD8\x27)
#define OMP_O_C_DMI_ALLOMORPHS mpP_dmi(\xD8\x28)
#define OMP_O_C_DMI_ATTRIBUTE_IDENTIFIER_LIST mpP_dmi(\xD8\x29)
#define OMP_O_C_DMI_ATTRIBUTE_LIST mpP_dmi(\xD8\x2A)
#define OMP_O_C_DMI_ATTRIBUTE_VALUE_CHANGE_DEFINITION mpP_dmi(\xD8\x2B)
#define OMP_O_C_DMI_ATTRIBUTE_VALUE_CHANGE_INFO mpP_dmi(\xD8\x2C)
#define OMP_O_C_DMI_AVAILABILITY_STATUS mpP_dmi(\xD8\x2D)
#define OMP_O_C_DMI_BACK_UP_DESTINATION_LIST mpP_dmi(\xD8\x2E)
#define OMP_O_C_DMI_BACK_UP_RELATIONSHIP_OBJECT mpP_dmi(\xD8\x2F)
#define OMP_O_C_DMI_CAPACITY_ALARM_THRESHOLD mpP_dmi(\xD8\x30)
#define OMP_O_C_DMI_CONTROL_STATUS mpP_dmi(\xD8\x31)
#define OMP_O_C_DMI_CORRELATED_NOTIFICATIONS mpP_dmi(\xD8\x32)
#define OMP_O_C_DMI_CORRELATED_NOTIFICATIONS-1 mpP_dmi(\xD8\x33)
#define OMP_O_C_DMI_COUNTER_THRESHOLD mpP_dmi(\xD8\x34)
#define OMP_O_C_DMI_DESTINATION mpP_dmi(\xD8\x35)
#define OMP_O_C_DMI_DOWN mpP_dmi(\xD8\x36)
#define OMP_O_C_DMI_GAUGE_THRESHOLD mpP_dmi(\xD8\x37)
#define OMP_O_C_DMI_GROUP_OBJECTS mpP_dmi(\xD8\x38)
#define OMP_O_C_DMI_INTERVALS_OF_DAY mpP_dmi(\xD8\x39)
#define OMP_O_C_DMI_MANAGEMENT_EXTENSION mpP_dmi(\xD8\x3A)
#define OMP_O_C_DMI_MONITORED_ATTRIBUTES mpP_dmi(\xD8\x3B)
#define OMP_O_C_DMI_MULTIPLE mpP_dmi(\xD8\x3C)
#define OMP_O_C_DMI_NOTIFY_THRESHOLD mpP_dmi(\xD8\x3D)
#define OMP_O_C_DMI_OBJECT_INFO mpP_dmi(\xD8\x3E)
#define OMP_O_C_DMI_OBSERVED_VALUE mpP_dmi(\xD8\x3F)
#define OMP_O_C_DMI_PACKAGES mpP_dmi(\xD8\x40)
#define OMP_O_C_DMI_PRIORITISED_OBJECT mpP_dmi(\xD8\x41)
#define OMP_O_C_DMI_PROBABLE_CAUSE mpP_dmi(\xD8\x42)
#define OMP_O_C_DMI_PROCEDURAL_STATUS mpP_dmi(\xD8\x43)
#define OMP_O_C_DMI_PROPOSED_REPAIR_ACTIONS mpP_dmi(\xD8\x44)
#define OMP_O_C_DMI_RELATIONSHIP_CHANGE_INFO mpP_dmi(\xD8\x45)
#define OMP_O_C_DMI_SECURITY_ALARM_DETECTOR mpP_dmi(\xD8\x46)
#define OMP_O_C_DMI_SECURITY_ALARM_INFO mpP_dmi(\xD8\x47)
#define OMP_O_C_DMI_SERVICE_USER mpP_dmi(\xD8\x48)
#define OMP_O_C_DMI_SETOF_ATTRIBUTE_VALUE_CHANGE_DEFINITION mpP_dmi(\xD8\x49)
#define OMP_O_C_DMI_SETOF_CORRELATED_NOTIFICATIONS mpP_dmi(\xD8\x4A)
#define OMP_O_C_DMI_SETOF_COUNTER_THRESHOLD mpP_dmi(\xD8\x4B)
#define OMP_O_C_DMI_SETOF_GAUGE_THRESHOLD mpP_dmi(\xD8\x4C)
#define OMP_O_C_DMI_SETOF_INTERVALS_OF_DAY mpP_dmi(\xD8\x4D)
#define OMP_O_C_DMI_SETOF_PRIORITISED_OBJECT mpP_dmi(\xD8\x4E)
#define OMP_O_C_DMI_SETOF_SUPPORTED_FEATURES mpP_dmi(\xD8\x4F)
#define OMP_O_C_DMI_SETOF_WEEK_MASK mpP_dmi(\xD8\x50)
#define OMP_O_C_DMI_SIMPLE_NAME_TYPE mpP_dmi(\xD8\x51)
#define OMP_O_C_DMI_SPECIFIC_IDENTIFIER mpP_dmi(\xD8\x52)
#define OMP_O_C_DMI_SPECIFIC_PROBLEMS mpP_dmi(\xD8\x53)
#define OMP_O_C_DMI_STATE_CHANGE_INFO mpP_dmi(\xD8\x54)
#define OMP_O_C_DMI_STOP_TIME mpP_dmi(\xD8\x55)
#define OMP_O_C_DMI_SUPPORTED_FEATURES mpP_dmi(\xD8\x56)
#define OMP_O_C_DMI_SYSTEM_ID mpP_dmi(\xD8\x57)
#define OMP_O_C_DMI_SYSTEM_TITLE mpP_dmi(\xD8\x58)
#define OMP_O_C_DMI_THRESHOLD_INFO mpP_dmi(\xD8\x59)
#define OMP_O_C_DMI_THRESHOLD_LEVEL_IND mpP_dmi(\xD8\x5A)
#define OMP_O_C_DMI_TIDE_MARK mpP_dmi(\xD8\x5B)
#define OMP_O_C_DMI_TIDE_MARK_INFO mpP_dmi(\xD8\x5C)
#define OMP_O_C_DMI_TIME24 mpP_dmi(\xD8\x5D)
#define OMP_O_C_DMI_UP mpP_dmi(\xD8\x5E)
#define OMP_O_C_DMI_WEEK_MASK mpP_dmi(\xD8\x5F)

```

```

/* Attribute Name Constants */
#define DMI_ADDITIONAL_INFORMATION 11301
#define DMI_ADDITIONAL_TEXT 11302
#define DMI_A_E_TITLE 11303
#define DMI_ALARM_STATUS 11304
#define DMI_APPLICATION 11305
#define DMI_ARM_TIME 11306
#define DMI_ATTRIBUTE 11307
#define DMI_ATTRIBUTE_ID 11308
#define DMI_ATTRIBUTE_IDENTIFIER_LIST 11309
#define DMI_ATTRIBUTE_LIST 11310
#define DMI_ATTRIBUTE_VALUE_CHANGE_DEFINITION 11311
#define DMI_AVAILABILITY_STATUS 11312
#define DMI_BACKED_UP_STATUS 11313
#define DMI_BACK_UP_OBJECT 11314
#define DMI_CAPACITY_ALARM_THRESHOLD 11315
#define DMI_COMPARISON_LEVEL 11316
#define DMI_CONTINUAL 11317
#define DMI_CONTROL_STATUS 11318
#define DMI_CORRELATED_NOTIFICATIONS 11319
#define DMI_COUNTER_THRESHOLD 11320
#define DMI_CURRENT_TIDE_MARK 11321
#define DMI_DAYS_OF_WEEK 11322
#define DMI_DETAILS 11323
#define DMI_DISTINGUISHED_NAME 11324
#define DMI_DOWN 11325
#define DMI_FEATURE_IDENTIFIER 11326
#define DMI_FEATURE_INFO 11327
#define DMI_GAUGE_THRESHOLD 11328
#define DMI_GLOBAL_VALUE 11329
#define DMI_HIGH 11330
#define DMI_HOUR 11331
#define DMI_IDENTIFIER 11332
#define DMI_INFORMATION 11333
#define DMI_INTEGER 11334
#define DMI_INTERVALS_OF_DAY 11335
#define DMI_INTERVAL_END 11336
#define DMI_INTERVAL_START 11337
#define DMI_LOCAL_VALUE 11338
#define DMI_LOW 11339
#define DMI_MANAGEMENT_EXTENSION 11340
#define DMI_MAX_TIDE_MARK 11341
#define DMI_MECHANISM 11342
#define DMI_MIN_TIDE_MARK 11343
#define DMI_MINUTE 11344
#define DMI_MONITORED_ATTRIBUTES 11345
#define DMI_MULTIPLE 11346
#define DMI_NAME 11347
#define DMI_NEW_ATTRIBUTE_VALUE 11348
#define DMI_NO_OBJECT 11349
#define DMI_NOTHING 11350
#define DMI_NOTIFICATION_IDENTIFIER 11351
#define DMI_NOTIFICATION_ON_OFF 11352
#define DMI_NOTIFY_HIGH 11353
#define DMI_NOTIFY_LOW 11354
#define DMI_NOTIFY_ON_OFF 11355
#define DMI_NUMBER 11356
#define DMI_OBJECT 11357
#define DMI_OBJECT_CLASS 11358
#define DMI_OBJECT_INSTANCE 11359
#define DMI_OBJECT_NAME 11360

```

```

#define DMI_OBSERVED_VALUE 11361
#define DMI_OFFSET_VALUE 11362
#define DMI_OID 11363
#define DMI_OLD_ATTRIBUTE_VALUE 11364
#define DMI_PACKAGES 11365
#define DMI_PERCEIVED_SEVERITY 11366
#define DMI_PREVIOUS_TIDE_MARK 11367
#define DMI_PRIORITISED_OBJECT 11368
#define DMI_PRIORITY 11369
#define DMI_PROBABLE_CAUSE 11370
#define DMI_PROCEDURAL_STATUS 11371
#define DMI_PROPOSED_REPAIR_ACTIONS 11372
#define DMI_REAL 11373
#define DMI_RELATIONSHIP_CHANGE_DEFINITION 11374
#define DMI_RESET_TIME 11375
#define DMI_SECURITY_ALARM_CAUSE 11376
#define DMI_SECURITY_ALARM_DETECTOR 11377
#define DMI_SECURITY_ALARM_SEVERITY 11378
#define DMI_SERVICE_PROVIDER 11379
#define DMI_SERVICE_USER 11380
#define DMI_SIGNIFICANCE 11381
#define DMI_SINGLE 11382
#define DMI_SOURCE_INDICATOR 11383
#define DMI_SOURCE_OBJECT_INST 11384
#define DMI_SPECIFIC 11385
#define DMI_SPECIFIC_IDEN6IFIER 11386
#define DMI_SPECIFIC_IDENTIFIER_INTEGER_2 11387
#define DMI_SPECIFIC_IDENTIFIER_OBJECT_IDENTIFIER_1 11388
#define DMI_SPECIFIC_PROBLEMS 11389
#define DMI_STATE_CHANGE_DEFINITION 11390
#define DMI_STRING 11391
#define DMI_SUPPORTED_FEATURES 11392
#define DMI_THRESHOLD 11393
#define DMI_THRESHOLD_INFO 11394
#define DMI_THRESHOLD_LEVEL 11395
#define DMI_TREND_INDICATION 11396
#define DMI_TRIGGERED_THRESHOLD 11397
#define DMI_UP 11398
#define DMI_WEEK_MASK 11399

/* Enumerations */

/* enum(AdministrativeState) */
#define DMI_ADMINISTRATIVE_STATE_LOCKED 0
#define DMI_ADMINISTRATIVE_STATE_UNLOCKED 1
#define DMI_ADMINISTRATIVE_STATE_SHUTTING_DOWN 2

/* enum(LogFullAction) */
#define DMI_LOG_FULL_ACTION_WRAP 0
#define DMI_LOG_FULL_ACTION_HALT 1

/* enum(OperationalState) */
#define DMI_OPERATIONAL_STATE_DISABLED 0
#define DMI_OPERATIONAL_STATE_ENABLED 1

/* enum(PerceivedSeverity) */
#define DMI_PERCEIVED_SEVERITY_INDETERMINATE 0
#define DMI_PERCEIVED_SEVERITY_CRITICAL 1
#define DMI_PERCEIVED_SEVERITY_MAJOR 2
#define DMI_PERCEIVED_SEVERITY_MINOR 3
#define DMI_PERCEIVED_SEVERITY_WARNING 4

```

```

#define DMI_PERCEIVED_SEVERITY_CLEARED 5

/* enum(SourceIndicator) */
#define DMI_SOURCE_INDICATOR_RESOURCE_OPERATION 0
#define DMI_SOURCE_INDICATOR_MANAGEMENT_OPERATION 1
#define DMI_SOURCE_INDICATOR_UNKNOWN 2

/* enum(TrendIndication) */
#define DMI_TREND_INDICATION_LESS_SEVERE 0
#define DMI_TREND_INDICATION_NO_CHANGE 1
#define DMI_TREND_INDICATION_MORE_SEVERE 2

/* enum(UsageState) */
#define DMI_USAGE_STATE_IDLE 0
#define DMI_USAGE_STATE_ACTIVE 1
#define DMI_USAGE_STATE_BUSY 2

/* Integer Lists */

/* integer(priority) */

#define DMI_PRIORITY_LOWEST 0
#define DMI_PRIORITY_HIGHEST 127

/* integer(alarm-Status) */
#define DMI_ALARM_STATUS_UNDER_REPAIR 0
#define DMI_ALARM_STATUS_CRITICAL 1
#define DMI_ALARM_STATUS_MAJOR 2
#define DMI_ALARM_STATUS_MINOR 3
#define DMI_ALARM_STATUS_ALARM_OUTSTANDING 4

/* integer(availability-Status) */
#define DMI_AVAILABILITY_STATUS_IN_TEST 0
#define DMI_AVAILABILITY_STATUS_FAILED 1
#define DMI_AVAILABILITY_STATUS_POWER_OFF 2
#define DMI_AVAILABILITY_STATUS_OFF_LINE 3
#define DMI_AVAILABILITY_STATUS_OFF_DUTY 4
#define DMI_AVAILABILITY_STATUS_DEPENDENCY 5
#define DMI_AVAILABILITY_STATUS_DEGRADED 6
#define DMI_AVAILABILITY_STATUS_NOT_INSTALLED 7
#define DMI_AVAILABILITY_STATUS_LOG_FULL 8

/* integer(control-Status) */
#define DMI_CONTROL_STATUS_SUBJECT_TO_TEST 0
#define DMI_CONTROL_STATUS_PART_OF_SERVICES_LOCKED 1
#define DMI_CONTROL_STATUS_RESERVED_FOR_TEST 2
#define DMI_CONTROL_STATUS_SUSPENDED 3

/* integer(max-Log-Size) */
#define DMI_MAX_LOG_SIZE_UNLIMITED 0

/* integer(procedural-Status) */
#define DMI_PROCEDURAL_STATUS_INITIALIZATION_REQUIRED 0
#define DMI_PROCEDURAL_STATUS_NOT_INITIALIZED 1
#define DMI_PROCEDURAL_STATUS_INITIALIZING 2
#define DMI_PROCEDURAL_STATUS_REPORTING 3
#define DMI_PROCEDURAL_STATUS_TERMINATING 4

/* integer(standby-Status) */
#define DMI_STANDBY_STATUS_HOT_STANDBY 0
#define DMI_STANDBY_STATUS_COLD_STANDBY 1

```

```

#define DMI_STANDBY_STATUS_PROVIDING_SERVICE                2

/* bitstring(daysOfWeek) */
#define DMI_DAYS_OF_WEEK_SUNDAY                          0
#define DMI_DAYS_OF_WEEK_MONDAY                          1
#define DMI_DAYS_OF_WEEK_TUESDAY                         2
#define DMI_DAYS_OF_WEEK_WEDNESDAY                      3
#define DMI_DAYS_OF_WEEK_THURSDAY                      4
#define DMI_DAYS_OF_WEEK_FRIDAY                        5
#define DMI_DAYS_OF_WEEK_SATURDAY                      6

/* objectIdentifier(probable-Cause) */
#define DMI_PROBABLE_CAUSE_ADAPTER_ERROR                  "\x59\x03\x02\x00\x00\x01"
#define DMI_PROBABLE_CAUSE_APPLICATION_SUBSYSTEM_FAILURE "\x59\x03\x02\x00\x00\x02"
#define DMI_PROBABLE_CAUSE_BANDWIDTH_REDUCED             "\x59\x03\x02\x00\x00\x03"
#define DMI_PROBABLE_CAUSE_CALL_ESTABLISHMENT_ERROR      "\x59\x03\x02\x00\x00\x04"
#define DMI_PROBABLE_CAUSE_COMMUNICATIONS_PROTOCOL_ERROR "\x59\x03\x02\x00\x00\x05"
#define DMI_PROBABLE_CAUSE_COMMUNICATIONS_SUBSYSTEM_FAILURE "\x59\x03\x02\x00\x00\x06"
#define DMI_PROBABLE_CAUSE_CONFIGURATION_OR_CUSTOMIZATION_ERROR "\x59\x03\x02\x00\x00\x07"
#define DMI_PROBABLE_CAUSE_CONGESTION                   "\x59\x03\x02\x00\x00\x08"
#define DMI_PROBABLE_CAUSE_CORRUPT_DATA                  "\x59\x03\x02\x00\x00\x09"
#define DMI_PROBABLE_CAUSE_CPU_CYCLES_LIMIT_EXCEEDED     "\x59\x03\x02\x00\x00\x0A"
#define DMI_PROBABLE_CAUSE_DATA_SET_OR_MODEM_ERROR       "\x59\x03\x02\x00\x00\x0B"
#define DMI_PROBABLE_CAUSE_DEGRADED_SIGNAL               "\x59\x03\x02\x00\x00\x0C"
#define DMI_PROBABLE_CAUSE_DTE_DCE_INTERFACE_ERROR      "\x59\x03\x02\x00\x00\x0D"
#define DMI_PROBABLE_CAUSE_ENCLOSURE_DOOR_OPEN           "\x59\x03\x02\x00\x00\x0E"
#define DMI_PROBABLE_CAUSE_EQUIPMENT_MALFUNCTION        "\x59\x03\x02\x00\x00\x0F"
#define DMI_PROBABLE_CAUSE_EXCESSIVE_VIBRATION          "\x59\x03\x02\x00\x00\x10"
#define DMI_PROBABLE_CAUSE_FILE_ERROR                    "\x59\x03\x02\x00\x00\x11"
#define DMI_PROBABLE_CAUSE_FIRE_DETECTED                 "\x59\x03\x02\x00\x00\x12"
#define DMI_PROBABLE_CAUSE_FLOOD_DETECTED               "\x59\x03\x02\x00\x00\x13"
#define DMI_PROBABLE_CAUSE_FRAMING_ERROR                 "\x59\x03\x02\x00\x00\x14"
#define DMI_PROBABLE_CAUSE_HEATING_OR_VENTILATION_
    OR_COOLING_SYSTEM_PROBLEM                            "\x59\x03\x02\x00\x00\x15"
#define DMI_PROBABLE_CAUSE_HUMIDITY_UNACCEPTABLE        "\x59\x03\x02\x00\x00\x16"
#define DMI_PROBABLE_CAUSE_INPUT_OUTPUT_DEVICE_ERROR    "\x59\x03\x02\x00\x00\x17"
#define DMI_PROBABLE_CAUSE_INPUT_DEVICE_ERROR           "\x59\x03\x02\x00\x00\x18"
#define DMI_PROBABLE_CAUSE_LAN_ERROR                     "\x59\x03\x02\x00\x00\x19"
#define DMI_PROBABLE_CAUSE_LEAK_DETECTED                 "\x59\x03\x02\x00\x00\x1A"
#define DMI_PROBABLE_CAUSE_LOCAL_NODE_TRANSMISSION_ERROR "\x59\x03\x02\x00\x00\x1B"
#define DMI_PROBABLE_CAUSE_LOSS_OF_FRAME                "\x59\x03\x02\x00\x00\x1C"
#define DMI_PROBABLE_CAUSE_LOSS_OF_SIGNAL                "\x59\x03\x02\x00\x00\x1D"
#define DMI_PROBABLE_CAUSE_MATERIAL_SUPPLY_EXHAUSTED    "\x59\x03\x02\x00\x00\x1E"
#define DMI_PROBABLE_CAUSE_MULTIPLEXER_PROBLEM          "\x59\x03\x02\x00\x00\x1F"
#define DMI_PROBABLE_CAUSE_OUT_OF_MEMORY                "\x59\x03\x02\x00\x00\x20"
#define DMI_PROBABLE_CAUSE_OUTPUT_DEVICE_ERROR          "\x59\x03\x02\x00\x00\x21"
#define DMI_PROBABLE_CAUSE_PERFORMANCE_DEGRADED         "\x59\x03\x02\x00\x00\x22"
#define DMI_PROBABLE_CAUSE_POWER_PROBLEM                 "\x59\x03\x02\x00\x00\x23"
#define DMI_PROBABLE_CAUSE_PRESSURE_UNACCEPTABLE        "\x59\x03\x02\x00\x00\x24"
#define DMI_PROBABLE_CAUSE_PROCESSOR_PROBLEM            "\x59\x03\x02\x00\x00\x25"
#define DMI_PROBABLE_CAUSE_PUMP_FAILURE                  "\x59\x03\x02\x00\x00\x26"
#define DMI_PROBABLE_CAUSE_QUEUE_SIZE_EXCEEDED          "\x59\x03\x02\x00\x00\x27"
#define DMI_PROBABLE_CAUSE_RECEIVE_FAILURE              "\x59\x03\x02\x00\x00\x28"
#define DMI_PROBABLE_CAUSE_RECEIVER_FAILURE             "\x59\x03\x02\x00\x00\x29"
#define DMI_PROBABLE_CAUSE_REMOTE_NODE_TRANSMISSION_ERROR "\x59\x03\x02\x00\x00\x2A"
#define DMI_PROBABLE_CAUSE_RESOURCE_AT_OR_NEARING_CAPACITY "\x59\x03\x02\x00\x00\x2B"
#define DMI_PROBABLE_CAUSE_RESPONSE_TIME_EXCESSIVE     "\x59\x03\x02\x00\x00\x2C"
#define DMI_PROBABLE_CAUSE_RETRANSMISSION_RATE_EXCESSIVE "\x59\x03\x02\x00\x00\x2D"
#define DMI_PROBABLE_CAUSE_SOFTWARE_ERROR               "\x59\x03\x02\x00\x00\x2E"

```

```

#define DMI_PROBABLE_CAUSE_SOFTWARE_PROGRAM_ABNORMALLY_
    TERMINATED                                "\x59\x03\x02\x00\x00\x2F"
#define DMI_PROBABLE_CAUSE_SOFTWARE_PROGRAM_ERROR
    "\x59\x03\x02\x00\x00\x30"
#define DMI_PROBABLE_CAUSE_STORAGE_CAPACITY_PROBLEM
    "\x59\x03\x02\x00\x00\x31"
#define DMI_PROBABLE_CAUSE_TEMPERATURE_UNACCEPTABLE
    "\x59\x03\x02\x00\x00\x32"
#define DMI_PROBABLE_CAUSE_THRESHOLD_CROSSED
    "\x59\x03\x02\x00\x00\x33"
#define DMI_PROBABLE_CAUSE_TIMING_PROBLEM
    "\x59\x03\x02\x00\x00\x34"
#define DMI_PROBABLE_CAUSE_TOXIC_LEAK_DETECTED
    "\x59\x03\x02\x00\x00\x35"
#define DMI_PROBABLE_CAUSE_TRANSMIT_FAILURE
    "\x59\x03\x02\x00\x00\x36"
#define DMI_PROBABLE_CAUSE_TRANSMITTER_FAILURE
    "\x59\x03\x02\x00\x00\x37"
#define DMI_PROBABLE_CAUSE_UNDERLYING_RESOURCE_UNAVAILABLE
    "\x59\x03\x02\x00\x00\x38"
#define DMI_PROBABLE_CAUSE_VERSION_MISMATCH
    "\x59\x03\x02\x00\x00\x39"

/* objectIdentifier(security-Alarm-Cause) */
#define DMI_SECURITY_ALARM_CAUSE_AUTHENTICATION_FAILURE
    "\x59\x03\x02\x00\x01\x01"
#define DMI_SECURITY_ALARM_CAUSE_BREACH_OF_CONFIDENTIALITY
    "\x59\x03\x02\x00\x01\x02"
#define DMI_SECURITY_ALARM_CAUSE_CABLE_TAMPER
    "\x59\x03\x02\x00\x01\x03"
#define DMI_SECURITY_ALARM_CAUSE_DELAYED_INFORMATION
    "\x59\x03\x02\x00\x01\x04"
#define DMI_SECURITY_ALARM_CAUSE_DENIAL_OF_SERVICE
    "\x59\x03\x02\x00\x01\x05"
#define DMI_SECURITY_ALARM_CAUSE_DUPLICATE_INFORMATION
    "\x59\x03\x02\x00\x01\x06"
#define DMI_SECURITY_ALARM_CAUSE_INFORMATION_MISSING
    "\x59\x03\x02\x00\x01\x07"
#define DMI_SECURITY_ALARM_CAUSE_INFORMATION_MODIFICATION_
    DETECTED
    "\x59\x03\x02\x00\x01\x08"
#define DMI_SECURITY_ALARM_CAUSE_INFORMATION_OUT_OF_SEQUENCE
    "\x59\x03\x02\x00\x01\x09"
#define DMI_SECURITY_ALARM_CAUSE_INTRUSION_DETECTION
    "\x59\x03\x02\x00\x01\x0A"
#define DMI_SECURITY_ALARM_CAUSE_KEY_EXPIRED
    "\x59\x03\x02\x00\x01\x0B"
#define DMI_SECURITY_ALARM_CAUSE_NON_REPUDIATION_FAILURE
    "\x59\x03\x02\x00\x01\x0C"
#define DMI_SECURITY_ALARM_CAUSE_OUT_OF_HOURS_ACTIVITY
    "\x59\x03\x02\x00\x01\x0D"
#define DMI_SECURITY_ALARM_CAUSE_OUT_OF_SERVICE
    "\x59\x03\x02\x00\x01\x0E"
#define DMI_SECURITY_ALARM_CAUSE_PROCEDURAL_ERROR
    "\x59\x03\x02\x00\x01\x0F"
#define DMI_SECURITY_ALARM_CAUSE_UNAUTHORIZED_ACCESS_ATTEMPTED
    "\x59\x03\x02\x00\x01\x10"
#define DMI_SECURITY_ALARM_CAUSE_UNEXPECTED_INFORMATION
    "\x59\x03\x02\x00\x01\x11"
#define DMI_SECURITY_ALARM_CAUSE_UNSPECIFIED_REASON
    "\x59\x03\x02\x00\x01\x12"

```

A.9 Workspace Content Package Output

This section provides the API Workspace Content Package Output which would result from applying the GDMO-XOM translation algorithm against the DMI information model.

```

SMI-PART2 {1.2.826.0.1050.6.4} BEGIN
  IMPORTS CMIS-Filter (2021),
  Attribute (2006),
  Base-Managed-Object-Id (2011),
  Attribute-Id (2008),
  Object-Instance (2051),
  Object-Class (2050),
  Event-Type-Id (2037)
  FROM OMP_O_MP_CMIS_PKG {1.2.826.0.1050.6.2};
  AE-Title (1005),
  DS-DN (1017)
  FROM OMP_O_MP_COMMON_PKG {1.2.826.0.1050.6.1};
  ATTRIBUTE VALUES
  Active-Destination {2.9.3.2.7.49} SYNTAX Destination;
  Additional-Information {2.9.3.2.7.6} SYNTAX Additional-Information;
  Additional-Text {2.9.3.2.7.7} SYNTAX GRAPHICSTRING;
  Administrative-State {2.9.3.2.7.31} SYNTAX ENUMERATION;
  Alarm-Status {2.9.3.2.7.32} SYNTAX Alarm-Status;
  Allomorphs {2.9.3.2.7.50} SYNTAX Allomorphs;
  Attribute-Identifier-List {2.9.3.2.7.8} SYNTAX Attribute-Identifier-List;
  Attribute-List {2.9.3.2.7.9} SYNTAX Attribute-List;
  Attribute-Value-Change-Definition {2.9.3.2.7.10} SYNTAX Setof-Attribute-Value-Change-Definition;
  Availability-Status {2.9.3.2.7.33} SYNTAX Availability-Status;
  Back-Up-Destination-List {2.9.3.2.7.51} SYNTAX Back-Up-Destination-List;
  Back-Up-Object {2.9.3.2.7.40} SYNTAX Back-Up-Relationship-Object;
  Backed-Up-Object {2.9.3.2.7.41} SYNTAX Back-Up-Relationship-Object;
  Backed-Up-Status {2.9.3.2.7.11} SYNTAX BOOLEAN;
  Capacity-Alarm-Threshold {2.9.3.2.7.52} SYNTAX Capacity-Alarm-Threshold;
  Confirmed-Mode {2.9.3.2.7.53} SYNTAX BOOLEAN;
  Control-Status {2.9.3.2.7.34} SYNTAX Control-Status;
  Correlated-Notifications {2.9.3.2.7.12} SYNTAX Setof-Correlated-Notifications;
  Corrupted-PD-Us-Received-Counter {2.9.3.2.7.72} SYNTAX INTEGER;
  Corrupted-PD-Us-Received-Threshold {2.9.3.2.7.89} SYNTAX Setof-Counter-Threshold;
  Current-Log-Size {2.9.3.2.7.54} SYNTAX INTEGER;
  Destination {2.9.3.2.7.55} SYNTAX Destination;
  Discriminator-Construct {2.9.3.2.7.56} SYNTAX CMIS-Filter;
  Discriminator-Id {2.9.3.2.7.1} SYNTAX Simple-Name-Type;
  Event-Time {2.9.3.2.7.13} SYNTAX GENERALIZEDTIME;
  Event-Type {2.9.3.2.7.14} SYNTAX Event-Type-Id;
  Incoming-Connection-Reject-Error-Counter {2.9.3.2.7.73} SYNTAX INTEGER;
  Incoming-Connection-Reject-Error-Threshold {2.9.3.2.7.90} SYNTAX Setof-Counter-Threshold;
  Incoming-Connection-Requests-Counter {2.9.3.2.7.74} SYNTAX INTEGER;
  Incoming-Connection-Requests-Threshold {2.9.3.2.7.91} SYNTAX Setof-Counter-Threshold;
  Incoming-Disconnect-Counter {2.9.3.2.7.75} SYNTAX INTEGER;
  Incoming-Disconnect-Error-Counter {2.9.3.2.7.76} SYNTAX INTEGER;
  Incoming-Disconnect-Error-Threshold {2.9.3.2.7.92} SYNTAX Setof-Counter-Threshold;
  Incoming-Protocol-Error-Counter {2.9.3.2.7.77} SYNTAX INTEGER;
  Incoming-Protocol-Error-Threshold {2.9.3.2.7.93} SYNTAX Setof-Counter-Threshold;
  Intervals-Of-Day {2.9.3.2.7.57} SYNTAX Setof-Intervals-Of-Day;
  Log-Full-Action {2.9.3.2.7.58} SYNTAX ENUMERATION;
  Log-Id {2.9.3.2.7.2} SYNTAX Simple-Name-Type;
  Log-Record-Id {2.9.3.2.7.3} SYNTAX Simple-Name-Type;
  Logging-Time {2.9.3.2.7.59} SYNTAX GENERALIZEDTIME;
  Managed-Object-Class {2.9.3.2.7.60} SYNTAX Object-Class;
  Managed-Object-Instance {2.9.3.2.7.61} SYNTAX Object-Instance;
  Max-Log-Size {2.9.3.2.7.62} SYNTAX INTEGER;
  Member {2.9.3.2.7.42} SYNTAX Group-Objects;
  Monitored-Attributes {2.9.3.2.7.15} SYNTAX Monitored-Attributes;

```

Name-Binding {2.9.3.2.7.63} SYNTAX OBJECTIDENTIFIER;
 Notification-Identifier {2.9.3.2.7.16} SYNTAX INTEGER;
 Number-Of-Records {2.9.3.2.7.64} SYNTAX INTEGER;
 Object-Class {2.9.3.2.7.65} SYNTAX Object-Class;
 Octets-Received-Counter {2.9.3.2.7.78} SYNTAX INTEGER;
 Octets-Received-Threshold {2.9.3.2.7.94} SYNTAX Setof-Counter-Threshold;
 Octets-Retransmitted-Error-Counter {2.9.3.2.7.79} SYNTAX INTEGER;
 Octets-Retransmitted-Threshold {2.9.3.2.7.95} SYNTAX Setof-Counter-Threshold;
 Octets-Sent-Counter {2.9.3.2.7.80} SYNTAX INTEGER;
 Octets-Sent-Threshold {2.9.3.2.7.96} SYNTAX Setof-Counter-Threshold;
 Operational-State {2.9.3.2.7.35} SYNTAX ENUMERATION;
 Outgoing-Connection-Reject-Error-Counter {2.9.3.2.7.81} SYNTAX INTEGER;
 Outgoing-Connection-Reject-Error-Threshold {2.9.3.2.7.97} SYNTAX Setof-Counter-Threshold;
 Outgoing-Connection-Requests-Counter {2.9.3.2.7.82} SYNTAX INTEGER;
 Outgoing-Connection-Requests-Threshold {2.9.3.2.7.98} SYNTAX Setof-Counter-Threshold;
 Outgoing-Disconnect-Counter {2.9.3.2.7.83} SYNTAX INTEGER;
 Outgoing-Disconnect-Error-Counter {2.9.3.2.7.84} SYNTAX INTEGER;
 Outgoing-Disconnect-Error-Threshold {2.9.3.2.7.99} SYNTAX Setof-Counter-Threshold;
 Outgoing-Protocol-Error-Counter {2.9.3.2.7.85} SYNTAX INTEGER;
 Outgoing-Protocol-Error-Threshold {2.9.3.2.7.100} SYNTAX Setof-Counter-Threshold;
 Owner {2.9.3.2.7.43} SYNTAX Group-Objects;
 Packages {2.9.3.2.7.66} SYNTAX Packages;
 PD-Us-Received-Counter {2.9.3.2.7.86} SYNTAX INTEGER;
 PD-Us-Received-Threshold {2.9.3.2.7.101} SYNTAX Setof-Counter-Threshold;
 PD-Us-Retransmitted-Error-Counter {2.9.3.2.7.87} SYNTAX INTEGER;
 PD-Us-Retransmitted-Error-Threshold {2.9.3.2.7.102} SYNTAX Setof-Counter-Threshold;
 PD-Us-Sent-Counter {2.9.3.2.7.88} SYNTAX INTEGER;
 PD-Us-Sent-Threshold {2.9.3.2.7.103} SYNTAX Setof-Counter-Threshold;
 Peer {2.9.3.2.7.44} SYNTAX Back-Up-Relationship-Object;
 Perceived-Severity {2.9.3.2.7.17} SYNTAX ENUMERATION;
 Primary {2.9.3.2.7.45} SYNTAX Setof-Prioritised-Object;
 Probable-Cause {2.9.3.2.7.18} SYNTAX Probable-Cause;
 Procedural-Status {2.9.3.2.7.36} SYNTAX Procedural-Status;
 Proposed-Repair-Actions {2.9.3.2.7.19} SYNTAX Proposed-Repair-Actions;
 Provider-Object {2.9.3.2.7.46} SYNTAX Setof-Prioritised-Object;
 Relationship-Change-Definition {2.9.3.2.7.20} SYNTAX Setof-Attribute-Value-Change-Definition;
 Scheduler-Name {2.9.3.2.7.67} SYNTAX Object-Instance;
 Secondary {2.9.3.2.7.47} SYNTAX Setof-Prioritised-Object;
 Security-Alarm-Cause {2.9.3.2.7.21} SYNTAX OBJECTIDENTIFIER;
 Security-Alarm-Detector {2.9.3.2.7.22} SYNTAX Security-Alarm-Detector;
 Security-Alarm-Severity {2.9.3.2.7.23} SYNTAX ENUMERATION;
 Service-Provider {2.9.3.2.7.24} SYNTAX Service-User;
 Service-User {2.9.3.2.7.25} SYNTAX Service-User;
 Source-Indicator {2.9.3.2.7.26} SYNTAX ENUMERATION;
 Specific-Problems {2.9.3.2.7.27} SYNTAX Specific-Problems;
 Standby-Status {2.9.3.2.7.37} SYNTAX INTEGER;
 Start-Time {2.9.3.2.7.68} SYNTAX GENERALIZEDTIME;
 State-Change-Definition {2.9.3.2.7.28} SYNTAX Setof-Attribute-Value-Change-Definition;
 Stop-Time {2.9.3.2.7.69} SYNTAX Stop-Time;
 Supported-Features {2.9.3.2.7.70} SYNTAX Setof-Supported-Features;
 System-Id {2.9.3.2.7.4} SYNTAX System-Id;
 System-Title {2.9.3.2.7.5} SYNTAX System-Title;
 Threshold-Info {2.9.3.2.7.29} SYNTAX Threshold-Info;
 Trend-Indication {2.9.3.2.7.30} SYNTAX ENUMERATION;
 Unknown-Status {2.9.3.2.7.38} SYNTAX BOOLEAN;
 Usage-State {2.9.3.2.7.39} SYNTAX ENUMERATION;
 User-Object {2.9.3.2.7.48} SYNTAX Setof-Prioritised-Object;
 Week-Mask {2.9.3.2.7.71} SYNTAX Setof-Week-Mask;

NOTIFICATIONS

Attribute-Value-Change {2.9.3.2.10.1} SYNTAX Attribute-Value-Change-Info;
 Communications-Alarm {2.9.3.2.10.2} SYNTAX Alarm-Info;
 Environmental-Alarm {2.9.3.2.10.3} SYNTAX Alarm-Info;
 Equipment-Alarm {2.9.3.2.10.4} SYNTAX Alarm-Info;
 Integrity-Violation {2.9.3.2.10.5} SYNTAX Security-Alarm-Info;

Object-Creation {2.9.3.2.10.6} SYNTAX Object-Info;
 Object-Deletion {2.9.3.2.10.7} SYNTAX Object-Info;
 Operational-Violation {2.9.3.2.10.8} SYNTAX Security-Alarm-Info;
 Physical-Violation {2.9.3.2.10.9} SYNTAX Security-Alarm-Info;
 Processing-Error-Alarm {2.9.3.2.10.10} SYNTAX Alarm-Info;
 Qualityof-Service-Alarm {2.9.3.2.10.11} SYNTAX Alarm-Info;
 Relationship-Change {2.9.3.2.10.12} SYNTAX Relationship-Change-Info;
 Security-Service-Or-Mechanism-Violation {2.9.3.2.10.13} SYNTAX Security-Alarm-Info;
 State-Change {2.9.3.2.10.14} SYNTAX State-Change-Info;
 Time-Domain-Violation {2.9.3.2.10.15} SYNTAX Security-Alarm-Info;

PARAMETERS

Miscellaneous-Error {2.9.3.2.5.1} SYNTAX NULL;

OBJECT DESCRIPTION

Additional-Information (11301) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 management-Extension (11340) SYNTAX Management-Extension MULTIVALUED;
 END

Alarm-Info (11302) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 probable-Cause (11370) SYNTAX Probable-Cause MANDATORY;
 specific-Problems (11389) SYNTAX Specific-Problems OPTIONAL WITH TAG [1] IMPLICIT;
 perceived-Severity (11366) SYNTAX ENUMERATION MANDATORY;
 backed-Up-Status (11313) SYNTAX BOOLEAN OPTIONAL;
 back-Up-Object (11314) SYNTAX Object-Instance OPTIONAL WITH TAG [2];
 trend-Indication (11396) SYNTAX ENUMERATION OPTIONAL WITH TAG [3] IMPLICIT;
 threshold-Info (11394) SYNTAX Threshold-Info OPTIONAL WITH TAG [4] IMPLICIT;
 notification-Identifier (11351) SYNTAX INTEGER OPTIONAL WITH TAG [5] IMPLICIT;
 correlated-Notifications (11319) SYNTAX Setof-Correlated-Notifications OPTIONAL WITH TAG [6] IMPLICIT;
 state-Change-Definition (11390)
 SYNTAX Setof-Attribute-Value-Change-Definition OPTIONAL WITH TAG [7] IMPLICIT;
 monitored-Attributes (11345) SYNTAX Monitored-Attributes OPTIONAL WITH TAG [8] IMPLICIT;
 proposed-Repair-Actions (11372) SYNTAX Proposed-Repair-Actions OPTIONAL WITH TAG [9] IMPLICIT;
 additional-Text (11302) SYNTAX GRAPHICSTRING OPTIONAL;
 additional-Information (11301) SYNTAX Additional-Information OPTIONAL WITH TAG [10] IMPLICIT;
 END

Alarm-Status (11303) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 alarm-Status (11304) SYNTAX INTEGER MULTIVALUED;
 END

Allomorphs (11304) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 object-Class (11358) SYNTAX Object-Class MULTIVALUED;
 END

Attribute-Identifier-List (11305) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 attribute-Id (11308) SYNTAX Attribute-Id MULTIVALUED;
 END

Attribute-List (11306) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 attribute (11307) SYNTAX Attribute MULTIVALUED;
 END

Attribute-Value-Change-Definition (11307) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 attribute-Id (11308) SYNTAX Attribute-Id MANDATORY;
 old-Attribute-Value (11364) SYNTAX ANYDEFINEDBY attribute-Id OPTIONAL WITH TAG [1];
 new-Attribute-Value (11348) SYNTAX ANYDEFINEDBY attribute-Id MANDATORY WITH TAG [2];
 END

Attribute-Value-Change-Info (11308) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 source-Indicator (11383) SYNTAX ENUMERATION OPTIONAL;
 attribute-Identifier-List (11309) SYNTAX Attribute-Identifier-List OPTIONAL WITH TAG [1] IMPLICIT;
 attribute-Value-Change-Definition (11311) SYNTAX Setof-Attribute-Value-Change-Definition MANDATORY;
 notification-Identifier (11351) SYNTAX INTEGER OPTIONAL;

correlated-Notifications (11319) SYNTAX Setof-Correlated-Notifications OPTIONAL WITH TAG [2] IMPLICIT;
additional-Text (11302) SYNTAX GRAPHICSTRING OPTIONAL;
additional-Information (11301) SYNTAX Additional-Information OPTIONAL WITH TAG [3] IMPLICIT;
END

Availability-Status (11309) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
availability-Status (11312) SYNTAX INTEGER MULTIVALUED;
END

Back-Up-Destination-List (11310) SEQUENCEOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
ae-Title (11303) SYNTAX AE-Title MULTIVALUED;
END

Back-Up-Relationship-Object (11311) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
object-Name (11360) SYNTAX Object-Instance OPTIONAL;
no-Object (11349) SYNTAX NULL OPTIONAL;
END

Capacity-Alarm-Threshold (11312) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
capacity-Alarm-Threshold (11315) SYNTAX INTEGER MULTIVALUED;
END

Control-Status (11313) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
control-Status (11318) SYNTAX INTEGER MULTIVALUED;
END

Correlated-Notifications (11314) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
correlated-Notifications (11319) SYNTAX Correlated-Notifications-1 MANDATORY;
source-Object-Inst (11384) SYNTAX Object-Instance OPTIONAL;
END

Correlated-Notifications-1 (11315) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
notification-Identifier (11351) SYNTAX INTEGER MULTIVALUED;
END

Counter-Threshold (11316) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
comparison-Level (11316) SYNTAX INTEGER MANDATORY;
offset-Value (11362) SYNTAX INTEGER MANDATORY;
notification-On-Off (11352) SYNTAX BOOLEAN MANDATORY;
END

Destination (11317) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
single (11382) SYNTAX AE-Title OPTIONAL;
multiple (11346) SYNTAX Multiple OPTIONAL;
END

Down (11318) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
high (11330) SYNTAX Observed-Value MANDATORY;
low (11339) SYNTAX Observed-Value MANDATORY;
END

Gauge-Threshold (11319) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
notify-Low (11354) SYNTAX Notify-Threshold MANDATORY;
notify-High (11353) SYNTAX Notify-Threshold MANDATORY;
END

Group-Objects (11320) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
object-Instance (11359) SYNTAX Object-Instance MULTIVALUED;
END

Intervals-Of-Day (11321) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
interval-Start (11337) SYNTAX Time24 MANDATORY;
interval-End (11336) SYNTAX Time24 MANDATORY;
END

Management-Extension (11322) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT

ATTRIBUTE BEGIN
 identifier (11332) SYNTAX OBJECTIDENTIFIER MANDATORY;
 significance (11381) SYNTAX BOOLEAN OPTIONAL WITH TAG [1] IMPLICIT;
 information (11333) SYNTAX ANYDEFINEDBY identifier MANDATORY WITH TAG [2];
 END
 Monitored-Attributes (11323) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 attribute (11307) SYNTAX Attribute MULTIVALUED;
 END
 Multiple (11324) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 ae-Title (11303) SYNTAX AE-Title MULTIVALUED;
 END
 Notify-Threshold (11325) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 threshold (11393) SYNTAX Observed-Value MANDATORY;
 notify-On-Off (11355) SYNTAX BOOLEAN MANDATORY;
 END
 Object-Info (11326) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 source-Indicator (11383) SYNTAX ENUMERATION OPTIONAL;
 attribute-List (11310) SYNTAX Attribute-List OPTIONAL;
 notification-Identifier (11351) SYNTAX INTEGER OPTIONAL;
 correlated-Notifications (11319) SYNTAX Setof-Correlated-Notifications OPTIONAL WITH TAG [1] IMPLICIT;
 additional-Text (11302) SYNTAX GRAPHICSTRING OPTIONAL;
 additional-Information (11301) SYNTAX Additional-Information OPTIONAL WITH TAG [2] IMPLICIT;
 END
 Observed-Value (11327) CHOICE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 integer (11334) SYNTAX INTEGER OPTIONAL;
 real (11373) SYNTAX REAL OPTIONAL;
 END
 Packages (11328) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 packages (11365) SYNTAX OBJECTIDENTIFIER MULTIVALUED;
 END
 Prioritised-Object (11329) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 object (11357) SYNTAX Object-Instance MANDATORY;
 priority (11369) SYNTAX INTEGER MANDATORY;
 END
 Probable-Cause (11330) CHOICE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 global-Value (11329) SYNTAX OBJECTIDENTIFIER OPTIONAL;
 local-Value (11338) SYNTAX INTEGER OPTIONAL;
 END
 Procedural-Status (11331) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 procedural-Status (11371) SYNTAX INTEGER MULTIVALUED;
 END
 Proposed-Repair-Actions (11332) SETOF CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 specific-Identifier (11385) SYNTAX Specific-Identifier MULTIVALUED;
 END
 Relationship-Change-Info (11333) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
 ATTRIBUTE BEGIN
 source-Indicator (11383) SYNTAX ENUMERATION OPTIONAL;
 attribute-Identifier-List (11309) SYNTAX Attribute-Identifier-List OPTIONAL WITH TAG [1] IMPLICIT;
 relationship-Change-Definition (11374) SYNTAX Setof-Attribute-Value-Change-Definition MANDATORY;
 notification-Identifier (11351) SYNTAX INTEGER OPTIONAL;
 correlated-Notifications (11319) SYNTAX Setof-Correlated-Notifications OPTIONAL WITH TAG [2] IMPLICIT;
 additional-Text (11302) SYNTAX GRAPHICSTRING OPTIONAL;
 additional-Information (11301) SYNTAX Additional-Information OPTIONAL WITH TAG [3] IMPLICIT;
 END
 Security-Alarm-Detector (11334) CHOICE CONCRETE DERIVED FROM OM_OBJECT

```

ATTRIBUTE BEGIN
mechanism (11342) SYNTAX OBJECTIDENTIFIER OPTIONAL WITH TAG [0] IMPLICIT;
object (11357) SYNTAX Object-Instance OPTIONAL WITH TAG [1];
application (11305) SYNTAX AE-title OPTIONAL WITH TAG [2] IMPLICIT;
END
Security-Alarm-Info (11335) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
security-Alarm-Cause (11376) SYNTAX OBJECTIDENTIFIER MANDATORY;
security-Alarm-Severity (11378) SYNTAX ENUMERATION MANDATORY;
security-Alarm-Detector (11377) SYNTAX Security-Alarm-Detector MANDATORY;
service-User (11380) SYNTAX Service-User MANDATORY;
service-Provider (11379) SYNTAX Service-User MANDATORY;
notification-Identifier (11351) SYNTAX INTEGER OPTIONAL;
correlated-Notifications (11319) SYNTAX Setof-Correlated-Notifications OPTIONAL WITH TAG [1] IMPLICIT;
additional-Text (11302) SYNTAX GRAPHICSTRING OPTIONAL;
additional-Information (11301) SYNTAX Additional-Information OPTIONAL WITH TAG [2] IMPLICIT;
END
Service-User (11336) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
identifier (11332) SYNTAX OBJECTIDENTIFIER MANDATORY;
details (11323) SYNTAX ANYDEFINEDBY identifier MANDATORY;
END
Setof-Attribute-Value-Change-Definition (11337) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
attribute-Value-Change-Definition (11311) SYNTAX Attribute-Value-Change-Definition MULTIVALUED;
END
Setof-Correlated-Notifications (11338) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
correlated-Notifications (11319) SYNTAX Correlated-Notifications MULTIVALUED;
END
Setof-Counter-Threshold (11339) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
counter-Threshold (11320) SYNTAX Counter-Threshold MULTIVALUED;
END
Setof-Gauge-Threshold (11340) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
gauge-Threshold (11328) SYNTAX Gauge-Threshold MULTIVALUED;
END
Setof-Intervals-Of-Day (11341) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
intervals-Of-Day (11335) SYNTAX Intervals-Of-Day MULTIVALUED;
END
Setof-Prioritised-Object (11342) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
prioritised-Object (11368) SYNTAX Prioritised-Object MULTIVALUED;
END
Setof-Supported-Features (11343) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
supported-Features (11392) SYNTAX Supported-Features MULTIVALUED;
END
Setof-Week-Mask (11344) SETOF CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
week-Mask (11399) SYNTAX Week-Mask MULTIVALUED;
END
Simple-Name-Type (11345) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
number (11356) SYNTAX INTEGER OPTIONAL;
string (11391) SYNTAX GRAPHICSTRING OPTIONAL;
END
Specific-Identifier (11346) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
specific-Identifier-OBJECTIDENTIFIER-1 (11388) SYNTAX OBJECTIDENTIFIER OPTIONAL;
specific-Identifier-INTEGGER-2 (11387) SYNTAX INTEGER OPTIONAL;
END
Specific-Problems (11347) SETOF CONCRETE DERIVED FROM OM_OBJECT

```

```

ATTRIBUTE BEGIN
specific-Identifier (11386) SYNTAX Specific-Identifier MULTIVALUED;
END
State-Change-Info (11348) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
source-Indicator (11383) SYNTAX ENUMERATION OPTIONAL;
attribute-Identifier-List (11309) SYNTAX Attribute-Identifier-List OPTIONAL WITH TAG [1] IMPLICIT;
state-Change-Definition (11390) SYNTAX Setof-Attribute-Value-Change-Definition MANDATORY;
notification-Identifier (11351) SYNTAX INTEGER OPTIONAL;
correlated-Notifications (11319) SYNTAX Setof-Correlated-Notifications OPTIONAL WITH TAG [2] IMPLICIT;
additional-Text (11302) SYNTAX GRAPHICSTRING OPTIONAL;
additional-Information (11301) SYNTAX Additional-Information OPTIONAL WITH TAG [3] IMPLICIT;
END
Stop-Time (11349) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
specific (11385) SYNTAX GENERALIZEDTIME OPTIONAL;
continual (11317) SYNTAX NULL OPTIONAL;
END
Supported-Features (11350) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
feature-Identifier (11326) SYNTAX OBJECTIDENTIFIER MANDATORY;
feature-Info (11327) SYNTAX ANYDEFINEDBY feature-Identifier MANDATORY;
END
System-Id (11351) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
name (11347) SYNTAX GRAPHICSTRING OPTIONAL;
number (11356) SYNTAX INTEGER OPTIONAL;
nothing (11350) SYNTAX NULL OPTIONAL;
END
System-Title (11352) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
distinguished-Name (11324) SYNTAX DS-DN OPTIONAL;
oid (11363) SYNTAX OBJECTIDENTIFIER OPTIONAL;
nothing (11350) SYNTAX NULL OPTIONAL;
END
Threshold-Info (11353) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
triggered-Threshold (11397) SYNTAX Attribute-Id MANDATORY;
observed-Value (11361) SYNTAX Observed-Value MANDATORY;
threshold-Level (11395) SYNTAX Threshold-Level-Ind OPTIONAL WITH TAG [1];
arm-Time (11306) SYNTAX GENERALIZEDTIME OPTIONAL WITH TAG [2] IMPLICIT;
END
Threshold-Level-Ind (11354) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
up (11398) SYNTAX Up OPTIONAL WITH TAG [1] IMPLICIT;
down (11325) SYNTAX Down OPTIONAL WITH TAG [2] IMPLICIT;
END
Tide-Mark (11355) CHOICE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
max-Tide-Mark (11341) SYNTAX Observed-Value OPTIONAL WITH TAG [0];
min-Tide-Mark (11343) SYNTAX Observed-Value OPTIONAL WITH TAG [1];
END
Tide-Mark-Info (11356) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
current-Tide-Mark (11321) SYNTAX Tide-Mark MANDATORY;
previous-Tide-Mark (11367) SYNTAX Tide-Mark MANDATORY;
reset-Time (11375) SYNTAX GENERALIZEDTIME MANDATORY;
END
Time24 (11357) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN
hour (11331) SYNTAX INTEGER MANDATORY;
minute (11344) SYNTAX INTEGER MANDATORY;
END
Up (11358) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT
ATTRIBUTE BEGIN

```

```
high (11330) SYNTAX Observed-Value MANDATORY;  
low (11339) SYNTAX Observed-Value OPTIONAL;  
END  
Week-Mask (11359) SEQUENCE CONCRETE DERIVED FROM OM_OBJECT  
ATTRIBUTE BEGIN  
days-Of-Week (11322) SYNTAX BITSTRING LENGTH 7 MANDATORY;  
intervals-Of-Day (11335) SYNTAX Setof-Intervals-Of-Day MANDATORY;  
END  
END
```

A.10 Encoding Definitions

This section provides the Encoding Definition Output which would result from applying the GDMO-XOM translation algorithm against the DMI information model.

OM Class	OM Class Type		
Additional-Information	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
management-Extension	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Alarm-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
probable-Cause	EXPLICIT	UNIVERSAL	-
specific-Problem	IMPLICIT	CONTEXT	1
perceived-Severity	IMPLICIT	UNIVERSAL	-
backed-Up-Status	IMPLICIT	UNIVERSAL	-
back-Up-Object	EXPLICIT	CONTEXT	2
trend-Indication	IMPLICIT	CONTEXT	3
threshold-Info	IMPLICIT	CONTEXT	4
notification-Identifier	IMPLICIT	CONTEXT	5
correlated-Notifications	IMPLICIT	CONTEXT	6
state-Change-Definition	IMPLICIT	CONTEXT	7
monitored-Attributes	IMPLICIT	CONTEXT	8
proposed-Repair-Actions	IMPLICIT	CONTEXT	9
additional-Text	IMPLICIT	UNIVERSAL	-
additional-Information	IMPLICIT	CONTEXT	10

OM Class	OM Class Type		
Alarm-Status	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
alarm-Status	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Allomorphs	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
object-Class	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Attribute-Identifier-List	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
attribute-Id	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Attribute-List	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
attribute	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Attribute-Value-Change-Definition	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
attribute-Id	IMPLICIT	UNIVERSAL	-
old-Attribute-Value	EXPLICIT	CONTEXT	1
new-Attribute-Value	EXPLICIT	CONTEXT	2

OM Class	OM Class Type		
Attribute-Value-Change-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
source-Indicator	IMPLICIT	UNIVERSAL	-
attribute-Identifier-List	IMPLICIT	CONTEXT	1
attribute-Value-Change-Definition	IMPLICIT	UNIVERSAL	-
notification-Identifier	IMPLICIT	UNIVERSAL	-
correlated-Notifications	IMPLICIT	CONTEXT	2
additional-Text	IMPLICIT	UNIVERSAL	-
additional-Information	IMPLICIT	CONTEXT	3

OM Class	OM Class Type		
Availability-Status	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
availability-Status	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Back-Up-Destination-List	SEQUENCE-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
a-E-Title	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Back-Up-Relationship-Object	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
object-Name	IMPLICIT	UNIVERSAL	-
no-Object	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Capacity-Alarm-Threshold	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
capacity-Alarm-Threshold	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Control-Status	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
control-Status	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Correlated-Notifications	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
correlated-Notifications	IMPLICIT	UNIVERSAL	-
source-Object-Inst	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Correlated-Notifications-1	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
notification-Identifier	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Counter-Threshold	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
comparison-Level	IMPLICIT	UNIVERSAL	-
offset-Value	IMPLICIT	UNIVERSAL	-
notification-On-Off	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Destination	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
single	IMPLICIT	UNIVERSAL	-
multiple	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Down	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
high	IMPLICIT	UNIVERSAL	-
low	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Gauge-Threshold	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
notify-Low	IMPLICIT	UNIVERSAL	-
notify-High	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Group-Objects	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
object-Instance	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Intervals-Of-Day	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
interval-Start	IMPLICIT	UNIVERSAL	-
interval-End	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Management-Extension	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
identifier	IMPLICIT	UNIVERSAL	-
significance	IMPLICIT	CONTEXT	1
information	EXPLICIT	CONTEXT	2

OM Class	OM Class Type		
Monitored-Attributes	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
attribute	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Multiple	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
ae-Title	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Notify-Threshold	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
threshold	IMPLICIT	UNIVERSAL	-
notify-On-Off	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Object-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
source-Indicator	IMPLICIT	UNIVERSAL	-
attribute-List	IMPLICIT	UNIVERSAL	-
notification-Identifier	IMPLICIT	UNIVERSAL	-
correlated-Notifications	IMPLICIT	CONTEXT	1
additional-Text	IMPLICIT	UNIVERSAL	-
additional-Information	IMPLICIT	CONTEXT	2

OM Class	OM Class Type		
Observed-Value	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
integer	IMPLICIT	UNIVERSAL	-
real	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Packages	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
packages	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Prioritised-Object	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
object	IMPLICIT	UNIVERSAL	-
priority	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Probable-Cause	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
global-Value	IMPLICIT	UNIVERSAL	-
local-Value	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Procedural-Status	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
procedural-Status	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Proposed-Repair-Action	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
specific-Identifier	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Relationship-Change-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
source-Indicator	IMPLICIT	UNIVERSAL	-
attribute-Identifier-List	IMPLICIT	CONTEXT	1
relationship-Change-Definition	IMPLICIT	UNIVERSAL	-
notification-Identifier	IMPLICIT	UNIVERSAL	-
correlated-Notifications	IMPLICIT	CONTEXT	2
additional-Text	IMPLICIT	UNIVERSAL	-
additional-Information	IMPLICIT	CONTEXT	3

OM Class	OM Class Type		
Security-Alarm-Detector	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
mechanism	IMPLICIT	CONTEXT	0
object	EXPLICIT	CONTEXT	1
application	IMPLICIT	CONTEXT	2

OM Class	OM Class Type		
Security-Alarm-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
security-Alarm-Cause	IMPLICIT	UNIVERSAL	-
security-Alarm-Severity	IMPLICIT	UNIVERSAL	-
security-Alarm-Detector	EXPLICIT	UNIVERSAL	-
service-User	IMPLICIT	UNIVERSAL	-
service-Provider	IMPLICIT	UNIVERSAL	-
notification-Identifier	IMPLICIT	UNIVERSAL	-
correlated-Notifications	IMPLICIT	CONTEXT	1
additional-Text	IMPLICIT	UNIVERSAL	-
additional-Information	IMPLICIT	CONTEXT	2

OM Class	OM Class Type		
Service-User	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
identifier	IMPLICIT	UNIVERSAL	-
details	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Attribute-Value-Change-Definition	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
attribute-Value-Change-Definition	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Correlated-Notifications	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
correlated-Notifications	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Counter-Threshold	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
counter-Threshold	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Gauge-Threshold	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
gauge-Threshold	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Intervals-Of-Day	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
intervals-Of-Day	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Prioritised-Object	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
prioritised-Object	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Supported-Features	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
supported-Features	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Setof-Week-Mask	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
week-Mask	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Simple-Name-Type	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
number	IMPLICIT	UNIVERSAL	-
string	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Specific-Identifier	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
specific-Identifier-OBJECTIDENTIFIER-1	IMPLICIT	UNIVERSAL	-
specific-Identifier-INTEGER-2	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Specific-Problems	SET-OF		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
specific-Identifier	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
State-Change-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
source-Indicator	IMPLICIT	UNIVERSAL	-
attribute-Identifier-List	IMPLICIT	CONTEXT	1
state-Change-Definition	IMPLICIT	UNIVERSAL	-
notification-Identifier	IMPLICIT	UNIVERSAL	-
correlated-Notifications	IMPLICIT	CONTEXT	2
additional-Text	IMPLICIT	UNIVERSAL	-
additional-Information	IMPLICIT	CONTEXT	3

OM Class	OM Class Type		
Stop-Time	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
specific	IMPLICIT	UNIVERSAL	-
continual	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Supported-Features	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
feature-Identifier	IMPLICIT	UNIVERSAL	-
feature-Info	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
System-Id	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
name	IMPLICIT	UNIVERSAL	-
number	IMPLICIT	UNIVERSAL	-
nothing	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
System-Title	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
distinguished-Name	IMPLICIT	UNIVERSAL	-
oid	IMPLICIT	UNIVERSAL	-
nothing	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Threshold-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
triggered-Threshold	IMPLICIT	UNIVERSAL	-
observed-Value	IMPLICIT	UNIVERSAL	-
threshold-Level	EXPLICIT	CONTEXT	1
arm-Time	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Threshold-Level-Ind	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
up	IMPLICIT	CONTEXT	1
down	IMPLICIT	CONTEXT	2

OM Class	OM Class Type		
Tide-Mark	CHOICE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
max-Tide-Mark	EXPLICIT	CONTEXT	0
min-Tide-Mark	EXPLICIT	CONTEXT	1

OM Class	OM Class Type		
Tide-Mark-Info	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
current-Tide-Mark	IMPLICIT	UNIVERSAL	-
previous-Tide-Mark	IMPLICIT	UNIVERSAL	-
reset-Time	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Time24	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
hour	IMPLICIT	UNIVERSAL	-
minute	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Up	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
high	IMPLICIT	UNIVERSAL	-
low	IMPLICIT	UNIVERSAL	-

OM Class	OM Class Type		
Week-Mask	SEQUENCE		
OM Attribute	Implicit/Explicit	Tag Type	Tag Value
days-Of-Week	IMPLICIT	UNIVERSAL	-
intervals-Of-Day	IMPLICIT	UNIVERSAL	-

Glossary

Abstract Syntax Notation One

A notation which both enables complicated types to be defined and also enables values of these types to be specified. See reference "ASN.1) .

argument

Information which is passed to a *function* or *operation/notification* and which specifies the details of the processing to be performed.

ASN.1

See *Abstract Syntax Notation One*.

Attribute - Service

A property of a particular type concerning a managed object and appearing in the definition of the managed object. An attribute has a value.

attribute - object

See *OM attribute*.

attribute syntax

A definition of the set of values which an *attribute* may assume. It includes the data type, in ASN.1, and, usually, one or more *matching rules* by which values may be compared.

attribute type - service

That component of an *attribute* which indicates the class of information given by that attribute. It is an *object identifier*, and so completely unique. or an *Integer*, which is unique but within a particular application context.

attribute type - object

Any of various categories into which the client dynamically groups values on the basis of their semantics. It is an integer, unique only within the *package*.

attribute value - service

A particular instance of the class of information indicated by an *attribute type*.

attribute value - object

An atomic information object.

attribute value assertion (AVA)

An assertion that a particular attribute of a managed object has a particular value. A proposition, which may be true, false, or undefined, concerning the value of an attribute of *managed object*.

attribute value syntax

See *Syntax (Object)*.

Basic Encoding Rules

A set of rules used to encode ASN.1 values as strings of octets.

BER

See *Basic Encoding Rules*.

DMI

Definition of Management Information.

GDMO

Guidelines for the Definition of Managed Objects: OSI Structure of Management Information, Part 4.

immediate subordinate

In the MIT, a *managed object instance* is an immediate subordinate of another if its *distinguished name* is formed by appending its RDN to the distinguished name of the other managed object instance.

immediate superior

In the MIT, a *managed object instance* is the immediate superior of another if its *distinguished name*, followed by the RDN of the other, forms the distinguished name of the other managed object instance.

implementation defined

The feature is not consistent across all implementations, and each implementation will provide documentation of its behaviour.

may

With respect to implementations, the feature is optional. Applications should not rely on the existence of the feature.

managed object

Anything in some 'world', generally resources in the world of telecommunications and information processing or some part thereof, which is identifiable (can be named), and which it is subject to management and thus of interest to become information of the MIB. A view of one or more resources; these resources may exist independently of management concerns, or may exist to support the management of other resources. A managed object may also represent a relationship between resources. A managed object is the abstracted view of such a resource that represents its properties as seen by and for the purposes of management.

managed object class

A generic classification shared by a set of similar managed objects that have similar properties and fulfil similar purposes.

managed object instance

A member of a managed object class, distinguishable by an identifier from other instances of that class.

Management Information Base (MIB)

The conceptual repository of management information. The complete set of information to which the Management Information Services provide access and which includes all of the pieces of information which can be read or manipulated using the operations of the Management Information Services. It is made up of *managed objects*. The MIB of an agent is the set of *managed objects* made visible by the agent.

Management Information Tree (MIT)

The MIB considered as a tree, expressing the containment relationships between the *managed objects*.

management interaction

A single management operation or a single notification or an identified set of logically related management operations and notifications during which the manager and agent role do not change.

management notification

The act of informing about an event which occurred in a managed object.

name

A construct that singles out a particular managed object from all other managed objects. A name must be unambiguous (that is, denote just one managed object instance), however it need not be unique (that is, be the only name which unambiguously denotes the managed object).

name binding

A naming rule which specifies allowed superior-subordinate pairs of managed object classes and which attribute is to be used in the relative distinguished name.

object

An object is a composite information object comprising zero or more *OM attributes* of different types.

object identifier

A value (distinguishable from all other such values) which is associated with an information object.

OM attribute

An OM attribute comprises one or more values of a particular type (and therefore syntax).

OM class

A static grouping of OM objects, within a specification, based on both their semantics and their form.

package

A specified group of related *OM classes*, denoted by an *Object Identifier*

should

With respect to implementations, the feature is recommended, but it is not a mandatory requirement. Applications should not rely on the existence of the feature.

With respect to applications, the word is used to give guidelines for recommended programming practice. These guidelines should be followed if maximum portability is desired.

subclass

A managed object class whose specification is derived from an existing object class specification (its superclass). 'Subclass' is a transitive relationship.

subordinate managed object

A managed object instance farther from the root in the naming tree (MIT), contained in a superior managed object and named within the scope of its superior managed object. In the MIT, a *managed object instance* is subordinate to another if its *distinguished name* includes that of the other as a prefix.

superior managed object

A managed object instance closer to the root in the naming tree (MIT), containing one or more subordinate managed objects. The superior managed objects is the scope for naming its subordinate managed objects.

In the MIT, a *managed object instance* is superior to another if its *distinguished name* is included as a prefix of the distinguished name of other. Each entry has exactly one immediate superior.

syntax - object

An OM syntax is any of various categories into which the OSI-Abstract-Data Manipulation specification statically groups values on the basis of their form. These categories are additional to the OM type of the value.

undefined

A feature is undefined if this document imposes no portability requirements on applications for erroneous program construct or erroneous data. Implementations *may* specify the result of using the feature, but such specifications are not guaranteed to be consistent across all implementations. That is, it is a programming error to use the feature, unless the particular implementation specifies the result. Note that an undefined result is completely unpredictable and may include abnormal program termination.

unspecified

A feature is unspecified if this document imposes no portability requirements on applications for correct program construct or erroneous data. Implementations *may* specify the result of using the feature, but such specifications are not guaranteed to be consistent across all implementations. That is, it is always permissible to use the feature, but the result is not known unless specified by the particular implementation.

will

The feature is required to be implemented and applications can rely on its existence.

workspace

A space in which *OM objects* of certain *OM classes* can be created, together with an implementation of the OSI-Abstract-Data Manipulation functions which supports those OM classes.

Index

<omp_dmi.h>.....	63	Management Information Base (MIB).....	128
Abstract Syntax Notation One.....	127	Management Information Tree (MIT)	128
Action template.....	41	management interaction	128
algorithm	1	management notification	128
architecture	3	may	128
architecture details	14	name	129
argument	127	name binding.....	129
ASN.1	2, 127	NMF.....	2
ASN.1 Module translation	46	notification information syntax.....	73
ASN.1 Syntax Module	9	Notification template.....	43
attribute	65	object	129
attribute - object	127	object identifier	63, 66, 129
Attribute - Service	127	OM attribute	129
attribute enumeration lists	94	OM class	129
Attribute Group template.....	39	OM package template.....	28
attribute syntax	127	OM Package template.....	63
attribute tables.....	77	output constraints	24
Attribute template	36	output specification	13
attribute type - object.....	127	package.....	1, 129
attribute type - service.....	127	package definitions	15
attribute value - object.....	127	package template	5
attribute value - service.....	127	parameter syntax	74
attribute value assertion (AVA)	127	Parameter template	33
attribute value syntax.....	70, 127	purpose of algorithm	2
authoritative output.....	3	scope of algorithm.....	3
Basic Encoding Rules.....	127	should	129
BER	2, 127	SMF.....	63
C header	13, 19, 100	structure of management information.....	63
C headers.....	63	subclass.....	129
class hierarchy.....	75	subordinate managed object	129
DMI	2, 63, 127	superior managed object.....	129
DMI as example	5	syntax - object.....	129
DMI contents package	63	translation rules	60
encoding definitions	13, 21, 117	undefined	130
event type.....	65	unspecified.....	130
GDMO	1, 128	will	130
immediate subordinate	128	workspace	130
immediate superior.....	128	workspace contents package	23
implementation defined.....	128	workspace contents package output.....	109
input assumptions.....	10	Workspace Contents Package translation	57
input requirements.....	5		
ISO GDMO Template.....	6		
managed object	128		
managed object class	128		
Managed Object Class template	31		
managed object instance	128		

