INTERNATIONAL STANDARD

ISO/IEC 10164-18

First edition
1997-07-01

Information technology — Open Systems Interconnection — Systems Management: Software management function

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Gestion-systèmes: Fonction de gestion de logiciel

Reference number
ISO/IEC 10164-18:1997(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10164-18 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open systems interconnection, data management and open distributed processing*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation X.744.

ISO/IEC 10164 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — Systems Management*:

— *Part 1: Object management function*

— *Part 2: State management function*

— *Part 3: Attributes for representing relationships*

— *Part 4: Alarm reporting function*

— *Part 5: Event report management function*

— *Part 6: Log control function*

— *Part 7: Security alarm reporting function*

— *Part 8: Security audit trail function*

— *Part 9: Objects and attributes for access control*

— *Part 10: Usage metering function for accounting purposes*

— *Part 11: Metric objects and attributes*

— *Part 12: Test management function*

— *Part 13: Summarization function*

— *Part 14: Confidence and diagnostic test categories*

— *Part 15: Scheduling function*

— *Part 16: Management knowledge management function*

— *Part 17: Change over function*

— *Part 18: Software management function*

— *Part 19: Management domain and management policy management functions*

— *Part 20: Time management function*

— *Part 21: Command sequencer*

— *Part 22: Response time monitoring function*

Annexes A to F form an integral part of this part of ISO/IEC 10164. Annexes G and H are for information only.

Introduction

Operations Systems (OS) Network Elements (NE) and other systems will occasionally be subject to software changes. These changes may include program code, such as a new version of some call processing related programme or revised routeing tables. The need within a network is to be able to manage software that can be changed or modified by some remote OS. In this context management can include: control over the downloading process for transfer of a software product from an OS to another system such as an OS or NE; control of installing downloaded software into the suite existing at the system; and the ability to check the version of software installed in a system. This Recommendation | International Standard provides a standard X.700 message based means to mange this software process to meet these needs. The method of software transfer between an OS and another system is not specified or constrained in any way by this Recommendation | International Standard but practically could include file transfer, electronic mail or postal services with floppy disks. The role of this Recommendation | International Standard in this transfer is to mange the start of delivery to another system by maintaining information about the start of a software delivery and relating this to actual receipt, resulting from a notification by the receiving system of successful delivery.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SYSTEMS MANAGEMENT: SOFTWARE MANAGEMENT FUNCTION

1 Scope

The Software Management Function includes management of a system for delivery of software and also management of software within a system.

There are two aspects of software that need to be considered separately. These two aspects can be described as the "dormant" view and the "active" view of software.

The dormant view of software is related to the data that is stored in a managed system and the way in which it is delivered and installed. In general, the data is stored information, such as data files and tables, but may also be files containing executable code. The scope of this Recommendation | International Standard includes the dormant view of software.

The active view of software is related to the management of resources that utilize the software. There is no real difference between this view and the normal view of management of resources. The scope of this Recommendation | International Standard does not include the active view of software. However, the relationship between managed objects representing resources that utilize software, and the managed objects representing the software that they are using (i.e. dormant view of software) is within the scope of this Recommendation | International Standard.

The scope of this Recommendation | International Standard includes:

- initiation of transfer of software;
- post transfer control of software;
- software activation (includes version update and patching);
- software de-activation;
- software reversion change;
- software validation;
- software enquiry;
- software backup;
- software restore.

NOTE 1 – Backup and restore are defined in a generic way so that they can apply to resources other than software.

The scope of this Recommendation | International Standard does not include:

- transfer mechanism of software;
- physical storage of software (mapping of software to physical file store, such as to a floppy disk, hard disk, etc.);
- formatting of software;
- naming of software products;
- sequencing of software management commands;
- software monitoring;
- management of processes running in a system.

NOTE 2 – The support for license charging activities (i.e. billing, accounting) and software tracing (i.e. tracing execution of software for debugging) is outside the scope of this Recommendation | International Standard and is open for further study.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and International Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: Conventions for the definition of OSI services.*
- CCITT Recommendation X.701 (1992) | ISO/IEC 10040:1992, *Information technology – Open Systems Interconnection – Systems management overview*[1)].
- CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1:1993, *Information technology – Open Systems Interconnection – Structure of management information: Management Information Model.*
- CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.*
- CCITT Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.*
- ITU-T Recommendation X.723 (1993) | ISO/IEC 10165-5:1994, *Information technology – Open Systems Interconnection – Structure of management information: Generic management information.*
- ITU-T Recommendation X.724 (1993) | ISO/IEC 10165-6:1994, *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management.*
- CCITT Recommendation X.730 (1992) | ISO/IEC 10164-1:1993, *Information technology – Open Systems Interconnection – Systems management: Object management function.*
- CCITT Recommendation X.731 (1992) | ISO/IEC 10164-2:1993, *Information technology – Open Systems Interconnection – Systems management: State management function.*
- CCITT Recommendation X.732 (1992) | ISO/IEC 10164-3:1993, *Information technology – Open Systems Interconnection – Systems management: Attributes for representing relationships.*
- ITU-T Recommendation X.738 (1993) | ISO/IEC 10164-13:1995, *Information technology – Open Systems Interconnection – Systems management: Summarization function.*
- ITU-T Recommendation X.739 (1993) | ISO/IEC 10164-11:1994, *Information technology – Open Systems Interconnection – Systems management: Metric objects and attributes.*
- CCITT Recommendation X.740 (1992) | ISO/IEC 10164-8:1993, *Information technology – Open Systems Interconnection – Systems management: Security audit trail function.*
- ITU-T Recommendation X.741 (1995) | ISO/IEC 10164-9:1995, *Information technology – Open Systems Interconnection – Systems Management: Objects and Attributes for Access Control.*
- ITU-T Recommendation X.742 (1995) | ISO/IEC 10164-10:1995, *Information technology – Open Systems Interconnection – Systems management: Usage metering function for accounting purposes.*
- ITU-T Recommendation X.745 (1993) | ISO/IEC 10164-12:1994, *Information technology – Open Systems Interconnection – Systems management: Test management function.*
- ITU-T Recommendation X.746 (1995) | ISO/IEC 10164-15:1995, *Information technology – Open Systems Interconnection – Systems management: Scheduling function.*

1) As amended by ITU-T Rec. X.701/Cor.2 (1995) | ISO/IEC 10040/Cor.2:1995.

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*

- CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

- ITU-T Recommendation X.291 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification.*

  ISO/IEC 9646-2:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 2: Abstract Test Suite specification.*

- ITU-T Recommendation X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements.*

  ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements.*

- CCITT Recommendation X.700 (1992), *Management framework for Open Systems Interconnection (OSI) for CCITT applications.*

  ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework.*

- CCITT Recommendation X.710 (1991), *Common management information service definition for CCITT Applications.*

  ISO/IEC 9595:1991, *Information technology – Open Systems Interconnection – Common management information service definition.*

- CCITT Recommendation X.711 (1991), *Common management information protocol specification for CCITT applications.*

  ISO/IEC 9596-1:1991, *Information technology – Open Systems Interconnection – Common management information protocol – Part 1: Specification.*

2.3 Additional references

- ITU-T Recommendation M.3100 (1995), *Generic network Information Model.*
- ITU-T Recommendation M.3101 (1995), *Managed object conformance statements for the "generic network Information Model".*
- ISO/IEC 15068-2[2], *Information technology – Portable Operating System Interface POSIX System Administration – Part 2: Software Administration (IEEE P1387.2).*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 Management framework definitions

This Recommendation | International Standard makes use of the following terms defined in CCITT Rec. X.700 and ISO/IEC 7498-4:

- managed object.

3.2 Systems management overview definitions

This Recommendation | International Standard makes use of the following terms defined in CCITT Rec. X.701 | ISO/IEC 10040:

a) managed object class;

---

[2] Currently at the stage of draft.

b) Management Information Conformance Statement (MICS);

c) Managed Object Conformance Statement (MOCS);

d) MICS proforma;

e) MOCS proforma;

f) notification.

3.3 CMIS definitions

This Recommendation | International Standard makes use of the following term defined in CCITT Rec. X.710 and ISO/IEC 9595:

– attribute.

3.4 Management information model definitions

This Recommendation | International Standard makes use of the following terms defined in CCITT Rec. X.720 | ISO/IEC 10165-1:

a) action;

b) behaviour;

c) name binding;

d) package;

e) superclass.

3.5 Guidelines for the definition of managed objects definitions

This Recommendation | International Standard makes use of the following term defined in CCITT Rec. X.722 | ISO/IEC 10165-4:

– template.

3.6 Implementation conformance statement proforma definitions

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.724 | ISO/IEC 10165-6:

a) Managed Relationship Conformance Statement (MRCS);

b) Management Conformance Summary (MCS);

c) MCS proforma;

d) MRCS proforma.

3.7 Additional definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.7.1 backup: The **backup** operations when applied to software managed object causes a copy of the underlying resources to be made. It has no direct effect on the original software resources.

NOTE – The **backup** operation may well be applicable to other classes of managed objects, however as far as this Recommendation | International Standard is concerned, only its use with software is considered.

3.7.2 deliver: **Delivery** of software is the process of getting the resources associated with a software managed object in place on the system together with the software managed object.

NOTE – The way in which the delivery of software is done, and in particular the choice of transfer mechanism, is outside the scope of this Recommendation | International Standard.

3.7.3 executable: **Executable software** represents software for which execution may be initiated. It may be possible to cause executable software to be executed by management command, however it may only be possible to execute software by local action.

3.7.4 install: **Installation** of software puts it into a form suitable for use as opposed to the form in which it may be transferred between systems (i.e. delivered). Once software is installed, it may be **utilized** by other managed objects.

NOTE – Installation may also involve setting up relationships between software managed objects.

3.7.5 **restore**: The **restore** operation may be used to replace the resources associated with a software managed object with a copy made by an earlier backup.

NOTE – As with **backup**, the **restore** operation may well be applicable to other classes of managed objects, however as far as this Recommendation | International Standard is concerned, only its use with software is considered.

3.7.6 **revert**: The **revert** operation reverses the effect of a previous **install** operation, or cause one or more applied patches to be removed.

3.7.7 **utilize**: **Utilization** is represented by relationships between some active processes and software resources. It represents the fact that the processes are using the software. Software may only be utilized when it is has an Operational state of ENABLED and an Administrative state of UNLOCKED. There may well be constraints as to how many things and what things may utilize a software resource. When software is utilized its Usage state will be either ACTIVE or BUSY.

NOTE – From a management point of view, the definition of these relationships is outside the scope of this Recommendation | International Standard.

3.7.8 **validate**: The **validate** operation is used to check the integrity of software resources.

NOTE – How the validation check is carried out is a local matter.

4 Symbols and abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

CMIP	Common Management Information Protocol
Conf	Confirmation
ICS	Implementation Conformance Statement
Id	Identifier
Ind	Indication
MCS	Management Conformance Summary
MICS	Management Information Conformance Statement
MOCS	Managed Object Conformance Statement
MRCS	Managed Relationship Conformance Statement
PICS	Protocol Implementation Conformance Statement
POSIX	Portable Operating System Interface
Req	Request
Rsp	Respond

5 Conventions

This Specification defines services for establishing and terminating backup relationship following the descriptive conventions defined in ITU-T Rec X.210 | ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model: Conventions for the definition of OSI services.*

The following notation is used in this Recommendation | International Standard's service parameter tables:

M The parameter is mandatory.

(=) The value of the parameter is equal to the value of the parameter in the column to the left.

U The use of the parameter is a Service-user option.

– The parameter is not present in the interaction described by the primitive concerned.

C The parameter is conditional.

P The parameter is subject to the constraints imposed by CCITT Rec. X.710 and ISO/IEC 9595.

6 Requirements for software management

Software Management must be able to satisfy the following requirements, subject to possible imposed controls and conditions:

a) be able to request delivery of software to a specific managed system;

b) be able to control the installation of software on a managed system, including the installation of patches (e.g. upgrades) and to revert back to a previous version of the software;

c) be able to initiate the execution of a software program;

d) be able to enquire as to the attributes of all software held on a managed system;

e) be able to create and delete software held on a managed system;

f) be able to validate software held on a managed system in order to check its integrity and to terminate validation;

g) be able to restrict use of software resources on a managed system for administrative purposes;

h) be able to back up a software item and to restore a previously backed up software item.

The model shall not preclude the use of logging, accounting, auditing, and license management at each state of this process.

In all cases the success or failure of the operation needs to be reported to the managing system.

Relationships exist between software managed objects that reflect the way in which software resources utilize other software resources. This relationship is known as "utilization", and is used to indicate which versions of software are to be used at any time. The management of this relationship is for further study.

7 Model for Software Management Function

The Software Management Function is primarily concerned with managing pieces of software. In this context software includes data, control information and executable instructions. The management view of software may be represented by a managed object of class "**Software Unit**"; management of software representing executable instructions has additional characteristics and so a second class. "**Executable Software Unit**" is defined (which is a sub-class of Software Unit).

Another aspect of software management is the management of delivery of software to the managed system. Software is not necessarily managed in the same units as it is delivered. For example, it is possible that a number of different software units may well be delivered together to a managed system (e.g. on a CD-ROM). Similarly, if a large software unit is to be delivered over a communications network, it may well be necessary to break it up into smaller parts for the purpose of delivery. Another case concerns the delivery of only the changes required to an existing software object, for example a patch. Management of delivery is done in terms of the **"Software Distributor"** managed object.

7.1 Software management functions

7.1.1 Create

A software unit may be created on a managed system using the Create operation. A software unit may be created either in the created or the delivered state, depending on behaviour. A software unit may be replicated or copied via a Create with Reference Object management operation with reference to that software unit managed object. A software unit may also be created as a result of the Delivery operation on some Software Distributor managed object or as a result of some local action. As a result of being created, a software unit managed object may emit an object creation notification.

7.1.2 Delete

A software unit may be deleted from a managed system using the Delete operation. A side effect of the Delete operation on a software unit may be the removal of associated underlying resources.

7.1.3 Deliver

Delivery of a coordinated set of software units may be requested. The result of delivery indicates the success or failure. Delivery is accomplished by sending the Deliver action to the Software Distributor object. The result of a Deliver action is that a copy of the target software items is delivered to the target system in the delivered state. A side effect may be the creation of one or more associated objects (e.g. software units).

Packaging of the software units and the choice of transfer mechanism is a local matter and outside the scope of this Recommendation | International Standard. For example, this information may be pre-configured or specified in the Deliver action along with any other associated information.

7.1.4 Execute program

A managing system may request the execution of executable software via the **execute program** operation. This Recommendation | International Standard does not specify how such an execution of the executable software may be subsequently managed, but merely provides a mechanism for initiating the execution of the executable software.

7.1.5 Get

It is possible to find out information about software (what software is present, what software is available for use, relationships between software, etc.) using the Get operation. If successful, the result of Get operation contains the information requested.

7.1.6 Install

Installation customizes the software for use. This can potentially require a significant amount of processing and time, and may involve checking that all parts of the target software version are present on the managed system (whether they have been delivered as part of an update or are already present) and assembling them ready for use. Installation is accomplished via the Install action directed to the Software Unit managed object, where it is an optional characteristic.

A patch is a modification to software and may be represented by a software managed object. Therefore, a patch may be delivered, installed, utilized, copied, etc., using software management.

A special case of Installation is that of a "patch", where it is the patch that is delivered, and the application of the patch is the installation and produces an updated version of the software ready for utilization. The updated version receives an updated version number for identification, delivered with the patch. The result of an installation indicates its success or failure.

When delivery of the software is complete, installation may commence. The installation may need to be coordinated, however such coordination is outside the scope of this Recommendation | International Standard. Examples of coordination include:

- completing delivery of several software items before another software item may be installed;
- checking that the current software is not in use before enabling a new one;
- checking that particular versions of other software are (not) installed;
- synchronizing installation with other open systems, providing for simultaneous but uncoupled installation on more than one open system, providing a way for installation on more than one open system to be bound together in a single unit of work, etc.

7.1.7 Revert

The Revert operation is used to cause an installed Software Unit managed object to revert back to being not installed or to cause one or more applied patches to be removed. The patches to be removed are identified by the file name or the Software Unit managed object instance id. The Revert operation is accomplished via an action directed to the Software Unit managed object, where it is an optional characteristic.

It may be necessary to maintain additional information regarding how a Software Unit managed object may be reverted. However, this is outside the scope of this Recommendation | International Standard.

7.1.8 Set administrative state

Once software has been customized for use (installed), the software may be made available for use via the Set administrative state to unlocked operation directed to the Software Unit managed objects. It is possible to remove specified software from a state of availability by setting the administrative state of the Software Unit managed object to shutting down or locked.

7.1.9 Terminate Validation

The validation of software may be terminated using the Terminate Validation operation. The Terminate Validation operation is accomplished via an action directed to the Software Unit managed object, where it is an optional characteristic. Validation can be terminiated prematurely because it could potentially take a long time.

7.1.10 Validation

The integrity of software may be checked using the Validate operation. It is also possible to validate that software previously delivered remains in a usable condition. The result of validation indicates whether the software was validated. The Validate operation is accomplished via an action directed to the Software Unit managed object, where it is an optional characteristic.

7.1.11 Notifications

All the operations applicable to a software unit may require that results be sent to the managing system along with the confirmation that the operation has been completed. It may also be necessary to notify another managing system of completion of one of these operations when the request for the operation was not originated from that managing system.

7.2 Other functions

The Backup and Restore functions are generic and are intended to be used for other managed objects in addition to the software objects in this Recommendation | International Standard.

7.2.1 Backup

The Backup operation may be used by a managing system to request performing a backup on a target object. The Backup operation when applied to software managed object causes a copy of the underlying resources to be made. It has no direct effect on the original software resources.

The Backup operation may well be applicable to other classes of managed objects, however as far as this Recommendation | International Standard is concerned, only its use with software is considered.

7.2.2 Restore

The Restore operation may be used by a managing system to request performing a restore of a previously backed up target object.

The Restore operation may well be applicable to other classes of managed objects, however as far as this Recommendation | International Standard is concerned, only its use with software is considered.

7.3 Software Unit Managed Object

7.3.1 Lifecycle

When a Software Unit Managed Object is first created it represents the resource that is prepared to accept delivery of a piece of software.

There are four mechanisms or operations by which a software unit managed object may be created:

1) Software may be replicated onto a target managed system without any operation (e.g. by local means).
2) Use of the Create operation to create a software unit in the created or delivered state.
3) Use of the Create with Reference Object operation, which references a source Software Unit managed object, and which is directed to a target destination in order to copy or replicate the source software.
4) Use of the Delivery operation directed to the Software Distributor managed object in the source system. This may create Software Unit managed objects in the target destination in the delivered state. The delivery operation applied to a Software Distributor managed object causes the software to be delivered to a target destination. When all the required software has been successfully delivered, a delivery result notification is emitted by the Software Distributor managed object to signify that delivery has been completed.

The Create and Create with Reference object operations cause the creation of a Software Unit managed object on the target destination. Success of creation would be indicated by the existence of a Software Unit managed object on the target destination (via a create response or an object creation notification).

If previously created but not delivered, the software unit managed object may emit a state change notification when delivery is completed, indicating that its state has changed from "created" to "delivered". A software unit managed object may also be instantiated in the delivered state.

Once the software is in the delivered state, the next stage is for the software to become installed. Installation involves customizing the software for use. For example, installation may involve the preparation and customizing of the software unit for an upgrade, perhaps by taking a copy of the current software resources and applying the changes to the copy. Installation may involve setting up configuration and dependency relationships between the software unit and other software objects.

Once software has been installed, it may be utilized by other managed objects. It may need to be made available for use by setting the administrative state to unlocked if it is in the locked state. This operation is implemented by using the replace operation on the Administrative State attribute defined in CCITT Rec. X.731 | ISO/IEC 10164-2.

Making the software unavailable for use is accomplished by administratively locking it (i.e. set administrative state to locked or shutting down). This prevents any new processes from using the software. If the administrative state is set to locked, all currently running processes using this software are aborted. If the admistrative state is set to shutting down, the currently running process are allowed to keep running but new processes are not allowed. When all the currently running processes using this software have completed execution, the administrative state becomes locked. This operation is implemented by using the replace operation on the Administrative State attribute defined in CCITT Rec. X.731 | ISO/IEC 10164-2.

The Revert operation may be used to reverse the effect of a previous install operation, or to cause an applied patch to be removed.

The Validate operation causes the software to be checked for its integrity, perhaps by invoking an algorithm generating a checksum, checking for viruses, etc. It may also be possible to cause the validation process to be prematurely terminated using the Terminate Validation operation.

Backing up software using the Backup operation is viewed as copying software, for example, for utilization should the current version fail. Backed up software may be restored using the Restore operation.

7.3.2 States of the Software Unit Managed Object

A software unit managed object can be in one of several lifecycle states depending on the last operation that was performed on it. The possible states (and their meanings) are :

- Created – The delivery of the software product is not completed, however some arbitrary resources on the managed system have been allocated to the software unit.
- Delivered – The software unit has been successfully delivered to the managed system.
- Installed – The software has been successfully installed on the managed system.

The Table 1 maps the states for a Software Unit managed object to the state and status values defined in CCITT Rec. X.731 | ISO/IEC 10164-2.

Table 1 – Software Unit Lifecycle States

Software Unit State	{Initialization Required} value of Procedural Status	{Not Installed} value of Availability Status
Created	Present	Present
Delivered	Absent	Present
Installed	Absent	Absent

For the Software Unit managed object the Administrative State, Operational State, Procedural Status and Availability Status are mandatory state attributes, while the Usage State is an optional attribute.

The created, delivered and installed states are mutually exclusive, that is, a software unit must be in only one of these states at a given time. Independent of the created, delivered and installed states are the validating and failed states.

Secondary states are the validating and failed states, which may be present in addition to the lifecycle states. The validating state maps to an Availability Status value which includes the value {In Test}. A software unit managed object may enter or exit the validating state independently of the values of its other states. For example, a software unit may be in the validating state while failed (see Table 2).

Table 2 – Software Unit Validating State

Software Unit State	{In Test} value of Availability Status
Validating	Present

A software unit in the created, delivered or installed state may also be in the failed state. The failed state maps to an Availability Status value including the value {Failed}. Specific causes for entering and exiting the failed state are a local matter. A software unit managed object may enter or exit the failed state independently of the values of its other states. For examples, the side effect of a VALIDATE operation may be a transition to the failed state while in the validating state, or a software unit may exit the failed state as a result of a revert operation (see Table 3).

Table 3 – Software Unit Failed State

Software Unit State	{Failed} value of Availability Status
Failed	Present

7.3.3 Operations on the Software Unit Managed Object

Associated with a software unit managed object are a number of operations which are used to make it change state.

These operations are :

- Backup – Causes the software to be backed up onto the target destination.
- Create – Causes a new software unit managed object to come into existence on the managed system.
- Delete – Causes the software unit managed object to be deleted on the managed and may additionally have the effect of deletion of associated resources.
- Install – Prepares the software for utilization.
- Restore – Causes backed up software to be restored from the target destination.
- Revert – Causes the application of a patch or an installation to be reverted.
- Terminate Validation – Causes the validation process of the software to be terminated.
- Validate – Checks the integrity of the software.

The Create and Delete operations map onto the standard Create and Delete operations defined in CCITT Rec. X.720 | ISO/IEC 10165-1, while the remainder of these map onto Actions. In addition, attribute values may be queried and modified using the get and replace operations defined in CCITT Rec. X.720 | ISO/IEC 10165-1.

Only the state transitions in Table 4 are permitted. Table 4 only addresses operations which can effect the states, other operations that do not cause a state change are excluded.

7.4 Executable Software managed object

The Executable Software managed object class is a subclass of the Software Unit managed object class with additional characteristics to describe its functionality as executable. Executable software represents software that may be executed in some way such as via local means or remotely via the execute operation. It may be possible to cause executable software to be executed by management command, however it may only be possible to execute software by local action.

Although out of scope in this Recommendation | International Standard, the model does not preclude subclasses of executable software from including information such as whether software can have single or multiple users, under what conditions software is active or busy, or whether a maximum number of users are allowed.

Table 4 – Software Unit State Transition Matrix

States	(Non-existent)	Created	Delivered	Installed
(Non-existent)	N/A	Create [a)], Create With Reference [a)] or local means	Create [a)], Create With Reference [a)], or Software Delivered	N/A
Created	Delete	–	Software Delivered	N/A
Delivered	Delete	N/A	–	Install or local means [a)]
Installed	Delete	N/A	Revert [a)]	Revert or – [a)]

a) Depending on object behaviour or local means
N/A State change not possible
– Any operation that does not cause a state change may be applicable (e.g. backup, validate, terminate validation, restore)

7.4.1 Additional states for the Executable Software Managed Object

The usage state, defined in CCITT Rec. X.731 | ISO/IEC 10164-2, reflects whether the executable software is currently being utilized.

For the Executable Software managed object class, the Usage State is a mandatory state attribute in addition to those mandated in the Software Unit managed object. In the Executable Software managed object class, the Usage State values of Idle, Active and Busy are permitted.

7.4.2 Additional operations for the Executable Software Managed Object

The following are operations on the Executable Software Managed Object in addition to those defined for the Software Unit Managed Object:

– Execute Program – This causes the software managed object to be executed.

The Execute Program operation is used to initiate the execution of software program represented by the Executable Software. The Executable Software must be installed in order for it to be executed.

The behaviour of the specific executable software instance or local means will determine the state of the executable software when it has been executed.

7.5 Software Distributor Managed Object

A Software Distributor managed object is a static object which represents delivery mechanism(s) of a managed system. It is a managed object which distributes software to the target managed system when it receives a deliver operation from the managing system. The parameters of the deliver operation may be used to indicate the set of software to be delivered, the target destination for delivery, and the choice of transfer mechanism. Although this object class may be used to initiate delivery via a variety of transfer mechanisms, it does not model any such transfer mechanisms. These models are left for specializations.

This managed object emits a notification with results of the distribution when the distribution is completed.

7.5.1 Operations on the Software Distributor Managed Object

The following operations associated with a Software Distributor managed object are a number of operations:

– Create – Causes a new Software Distributor managed object to come into existence.

- Deliver – Causes the Software Distributor managed object to cause creation of the specified software (by a method out of scope of this Recommendation | International Standard) onto the target managed system and causes any associated resources which are to be associated with those software unit managed objects to be created onto the target managed as a side effect.
- Delete – Causes the Software Distributor managed object to be deleted on the managed system.

7.5.2 States of the Software Distributor Managed Object

A Software Distributor managed object may be in one of the following states depending on the last operation that was performed on it. The possible states are Created and Non-existent (see Table 5).

Table 5 – State transitions of software distributor managed object

Input State/Output State	Non-existent	Created
Non-existent	–	Create
Created	Delete	Deliver

7.6 Relationships

A number of relationships between software managed objects and also between software managed objects and other managed objects have been identified. These are:

- Dependency – This relationship may be used to model the fact that one software managed object is dependent in some way on the presence of another software managed object. Such a relationship can be used to model patching.
- Configuration – This relationship may be used to model the fact that one software unit managed object may affect the behaviour of another software unit managed object. For example an additional font could be modelled as a configuration relationship. This relationship can also be used to model enhancements and patching.
- Utilisation – This relationship may be used to show what other managed objects utilize software managed objects. Such other managed objects are likely to be those representing processes running on the managed system.

Failure of a managed object may cause any managed objects which are dependent upon it to also fail. However it will not cause managed objects with a configuration relationship to fail (though their behaviour may change). The detection of faulty software may result in the requirement to deliver and install a new copy of the software or to deliver and install an updated version.

However specification of any of these relationships is outside the scope of this Recommendation | International Standard, as the relationships are application dependent.

8 Generic definitions

8.1 Managed objects

8.1.1 Software

The Software object class is a class of managed objects that represents logical information stored in a managed element, including programs and data tables.

This managed object class has the following mandatory characteristics:

- softwareID.

This managed object class has the following conditional packages (with the following characteristics):

- createDeleteNotificationsPackage (objectCreation and objectDeletion notifications);
- attributeValueChangeNotificationPackage (attributeValueChange notification);
- stateChangeNotificationPackage (stateChange notification);

- administrativeOperationalStatesPackage (administrativeState and operationalState attributes);
- affectedObjectListPackage (affectedObjectList attribute);
- softwareProcessingErrorAlarmPackage [alarmStatus attribute (read only) and processingErrorAlarm notification];
- userLabelPackage (userLabel attribute);
- vendorNamePackage (vendorName attribute);
- versionPackage (version attribute);
- currentProblemListPackage (currentProblemList attribute).

This managed object class is defined in Recommendation M.3100.

8.1.2 Software Unit

The software Unit object class is a class of managed objects that provide administrable information associated with software (whether it be in the form of an executable file, such as program software, or a non-executable file, such as a data or cross-connect mapping table). It is a subclass of Recommendation M.3100 software object class.

This managed object class has the following additional mandatory characteristics:

- administrativeState attribute;
- availabilityStatus attribute;
- operationalState attribute;
- proceduralStatus attribute;
- processingErrorAlarm notification with alarmEffectOnServiceParameter parameter.

This managed object class has the following additional conditional packages (with the following characteristics):

- appliedPatchPackage (appliedPatches attribute);
- checkSumPackage (checkSum attribute);
- fileInformationPackage (dateOfCreation, identityOfCreator, dateOfLastModification, identityOfLastModifier, dateDelivered, and dateInstalled attributes);
- filePackage (fileType, fileLocation, and fileSize attributes);
- informationAutoBackupPackage (autoBackupReport notification, futureAutoBackupTriggerThreshold and futureAutoBackupDestination attributes);
- informationAutoRestorePackage (autoRestoreReport notification, futureAutoRestoreSource and futureAutoRestoreAllowed attributes);
- informationBackupPackage (backup action, lastBackuptime and lastBackupDestination attributes);
- informationRestorePackage (restore action, lastRestoreTime and lastRestoreSource attributes);
- installPackage (install action);
- noteFieldPackage (noteField attribute);
- revertPackage (revert action);
- terminateValidationPackage (terminateValidation action);
- usageStatePackage (usageState attribute);
- validationPackage (validate action).

8.1.3 Executable Software

The executableSoftware object class is a class of managed objects that provide administrable information associated with an executable program in the managed system. This object class is a subclass of the Software Unit managed object class. The actual executable program (that may consist of code segments with or without data segments, etc.) may be in a non-standard, machine-dependent format that is generally unreadable by managing system.

An action called executeProgram (defined below) can be used to execute the program represented by the executableSoftware object instance. The usageState attribute is used to indicate if there are any active executions of the program. The value of idle means that there are no active executions. What the values of active and busy mean are implementation specific and are out of the scope of this Recommendation | International Standard. For example, the value of active could mean that there exist some running execution, while the value of busy means that the program has reached its maximum capacity and any further request of execution (by using the executeProgram action or by some other local means) may be denied or queued for later execution. Specializations may want to include a threshold for the number of users required to reach maximum capacity.

This managed object class has the following additional mandatory characteristics:

- usageState attribute.

This managed object class has the following additional conditional packages (with the following characteristics):

- executeProgramPackage (executeProgram action).

8.1.4 Software Distributor

A Software Distributor managed object is a static object which represents delivery mechanism(s) of a managed system. It is a managed object which distributes specified software to a specified target managed system when it receives a deliver operation from the managing system. The parameters of the deliver operation used indicate the set of software to be delivered, the target destination for delivery, and the choice of transfer mechanism. Although this object class may be used to initiate delivery via a variety of transfer mechanisms, it does not model any such transfer mechanisms. These models are left for specializations.

This managed object emits a notification with results of the distribution when the distribution is completed. The stateChangeNotification defined in CCITT Rec. X.721 | ISO/IEC 10165-2 shall be emitted if the value of the administrative state or operational state changes. The Software Distributor managed object class is a subclass of the top managed object class defined in CCITT Rec. X.721 | ISO/IEC 10165-2.

This managed object class has the following mandatory characteristics:

- administrativeState attribute;
- operationalState attribute;
- softwareDistributorId;
- deliver action;
- deliverResultNotification notification;
- objectCreation notification;
- objectDeletion notification;
- stateChange notification.

8.2 Attributes

8.2.1 appliedPatches

This attribute identifies the patches that have been applied to and still exist in the software unit which is represented by the softwareUnit object instance. Patches are updates to software. The value of this attribute is read-only and automatically updated when a patch is performed on the software. The syntax of this attribute is sequence of patch identifiers, where a patch identifier is a choice of object instance (if the patch is represented by a software unit managed object) or graphic string (if the patch is not represented by a software unit managed object).

8.2.2 checkSum

This attribute identifies the checksum of the software information represented by the softwareUnit object instance.

8.2.3 dateDelivered

This attribute identifies the time that the information represented by the softwareUnit object instance was delivered to the managed system. Valid values for this attribute are ASN.1 GeneralizedTime or NULL if the information has not been delivered.

8.2.4 dateInstalled

This attribute identifies the time that the information represented by the softwareUnit object instance was installed. Valid values for this attribute are ASN.1 GeneralizedTime or NULL if the information has not been installed.

8.2.5 dateOfCreation

This attribute indicates the time of creation of the software unit managed object. The syntax of this attribute is of ASN.1 GeneralizedTime type. The mandatory initial value of this attribute is the actual time (available through local means) of creation of the object instance.

8.2.6 dateOfLastModification

This attribute identifies the time of the last, or most recent modification (e.g. patching, reverting, installing, delivery) to the information represented by the software. Valid values for this attribute are ASN.1 GeneralizedTime or NULL if the information has not been modified.

NOTE – When last modification is a revert, the revert time becomes the date of last modification.

8.2.7 fileLocation

This attribute specifies the full address (either logical or physical) of the software file or files which are represented by a softwareUnit object. The syntax of this attribute is a choice of set of graphic strings or null. The format of an address is implementation-dependent, conforming to the file-addressing conventions of the particular managed system in question so the address is represented by a graphic string in syntax. A null value of this attribute indicates that the information to which the softwareUnit managed object applies has not yet been delivered to the managed system.

8.2.8 fileSize

This attribute indicates the size of the softwareUnit managed object.

8.2.9 fileType

This attribute indicates the type of the softwareUnit. Possible softwareUnit types are unstructured binary file (e.g. executable file), unstructured text file (e.g. non-executable file), and block special file, etc.

8.2.10 futureAutoBackupDestination

This attribute specifies the destination to which the information represented by this object instance will be backup. The backup criteria is defined in the futureAutoBackupTriggerThreshold attribute of the object instance. The destination can be another object instance of the same object class which exists in the same local managed system, a remote open system (by using a particular file transfer protocol, e.g. FTAM), or the managing system (by in-line using the autoBackupReport notification).

8.2.11 futureAutoBackupTriggerThreshold

This attribute specifies the threshold that will trigger an automatic backup for the information represented by the object instance. The threshold is defined as the number of time that the information has been modified. Once the information has been modified for that number of times, an automatic backup will be performed. The backup destination is specified in the attribute futureAutomaticBackupDestination. Such backups are carried out in addition to other scheduled periodic backup. At the completion of the automatic backup, an autoBackupReport notification shall be emitted from the object.

8.2.12 futureAutoRestoreAllowed

This attribute specifies whether automatic restore of the information represented by this managed object instance is allowed. The syntax of this attribute is of ASN.1 BOOLEAN type with the value TRUE meaning allowed, and FALSE meaning not allowed. The criteria that trigger automatic information restore are system specific.

8.2.13 futureAutoRestoreSource

This attribute specifies the source of the information to be restored to the information represented by the managed object instance. The source is either a local managed object or a remote system. The criteria that trigger an automatic restore of information are system specific.

8.2.14 identityOfCreator

This attribute identifies the entity that creates the managed object.

8.2.15 identityOfLastModifier

This attribute identifies the last, or most recent, modifier of the information represented by the softwareUnit object instance.

8.2.16 lastBackupDestination

This attribute identifies the destination, if exists, to which the information represented by the managed object is backed up.

8.2.17 lastBackupTime

This attribute identifies the time of the last backup on the information represented by the managed object instance. Valid values for this attribute are ASN.1 GeneralizedTime or NULL (if no backup has been performed on the information).

8.2.18 lastRestoreSource

This attribute identifies the source, if exists, from which the information represented by the managed object is restored.

8.2.19 lastRestoreTime

This attribute identifies the time of the last restore on the information represented by the managed object instance. Valid values for this attribute are ASN.1 GeneralizedTime or NULL (if no restore has been performed on the information).

8.2.20 noteField

This attribute contains any information or comments associated with the managed object, including any specific installation instructions, startup parameters and values, information necessary to activate features of the managed object, etc.

8.2.21 softwareDistributorId

The Software Distributor Id is used to identify instances of the software distributor managed object class.

8.3 Actions

The set of generic action parameters and semantics defined by this Recommendation | International Standard provide the detail for the following general parameters of the M-ACTION service defined by CCITT Rec. X.710 and ISO/IEC 9595:

- action type;
- action information;
- action reply.

8.3.1 backup

The backup service is used by a managing system to request performing a backup on the information represented by the target object instance (i.e. managed object representing the software being backed up.

This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate backup. The Action Information parameter shall indicate the destination to which the information will be backed up. Possible destination are:

- A local managed object – In this case, the backup operation will be performed internally in the managed system.
- The managing system from which this action is sent – In this case, a copy of the backup information will be sent in-line in the Action Reply.
- A remote system – In this case, the backup information will be transferred off-line to the remote system by some local means.

The Action Reply parameter shall indicate the result of the backup. For local or off-line backup, a NULL value indicates the backup is success. For in-line backup, the backup information will be included in the Action Reply parameter.

8.3.2 deliver

The deliver service is used by a managing system to request distribution of a software or a set of software. The deliver action information identifies the software that is to be distributed. A number of software units may be delivered together to a managed system. The result of successful completion is that the software to be distributed is copied to the target system; this may have the effect of softwareUnit objects being created.

This service used the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate deliver.

8.3.3 execute program

The execute program service is used by a managing system to initiate the execution of a program represented by an executable software object.

This service used the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate execute program.

8.3.4 install

The install service is used by a managing system to install a delivered software unit object. If the software unit cannot be installed due an invalid state condition an operationStateMismatch specific error is returned.

This service used the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate install.

8.3.5 restore

The restore service is used by a managing system to request performing a restore on the information represented by the target object instance. This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate restore. The Action Information parameter shall indicate the source from which the information will be restored. Possible sources are:

- A local managed object – In this case, the restore operation will be performed internally in the managed system.
- The managing system from which this action is sent – In this case, a copy of the restore information will be sent in-line in the Action Information parameter.
- A remote system – In this case, the restore information will be transferred off-line from the remote system by using some local means.

This action is always confirmed.

8.3.6 revert

The revert service is used by a managing system (e.g. OS) to instruct a managed system to revert an applied patch or set of patches of the software represented by the software unit managed object.

The revert service will automatically return the value of the appliedPatches attribute of the software unit object instance to which the service is directed. This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate revert. The Action Information parameter shall identify the patch or patches to be reverted of the software unit object instance. Each patch identifier is a choice of a system specific identifier (Printable String) or a software unit object instance (Object Instance).

8.3.7 terminateValidation

The terminateValidation service is used by a managing system to abort a currently active (running) validation. A pre-condition for this operation is that the object receiving the request is currently validating, otherwise the action reply indicates that there was no active validation process to be terminated.

Two termination modes are provided for terminating a validation, namely, cancel-mode and truncated-mode. For the truncate-mode, the validation will be terminated and the result of the partially completed validation will be returned in the Action Reply parameter of the terminateValidation M-ACTION. For the cancel-mode, the validation will be terminated and the result of the partially completed validation will be discarded.

This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate terminatevalidation. The Action Information parameter shall indicate the mode of the termination, i.e. cancel-mode or truncate-mode. The Action Reply parameter shall indicate the result of the termination, i.e. validation terminated (for cancel-mode), validation terminated with partial result (for truncate-mode), or no active validation to be aborted.

8.3.8 validate

The validate service is used by a managing system to request performing a validation on the information represented by the softwareUnit object instance. This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Action Type parameter shall indicate validate. The Action Information parameter shall indicate the type of validation to be performed (using a set of management extensions). The Action Reply parameter shall indicate the result of the validation. If the validation process is terminated (by using the terminate validation M- ACTION) before the normal completion of the validation, then the value 'terminated' (with ASN.1 type NULL) shall be returned in the Action Reply of the validate M-ACTION. If the termination mode of the terminate validation M-ACTION is 'truncate', then the result of the truncated validation (i.e. the part of the validation that already has completed), should be return in the Action Reply parameter of the terminate validation M-ACTION.

8.4 Notifications

The set of generic notifications, parameters and semantics defined by this Recommendation | International Standard provide the detail for the following general parameters of the M-EVENT-REPORT service as defined by CCITT Rec. X.710 and ISO/IEC 9595;

- event type;
- event information;
- event reply.

8.4.1 autoBackupReport

The autoBackupReport notification is emitted to report an automatic backup of the information represented by this object.

The automatic backup criteria and backup destination are specified in the futureAutoBackupTriggerThreshold and futureAutoBackupDestination attributes of the object respectively.

The backup destination may be local (i.e. backup to another object of the same class within the local managed system), the managing system, or off-line to a remote system by using a particular file transfer protocol (e.g. FTAM). For local and off-line backup, the result of the backup, i.e. success or failure, will be reported in this notification. For backup to the managing system, the backup information will be in-line included in the notification as a component of the Event Information parameter.

This service uses the M-EVENT-REPORT service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Event Type parameter shall indicate auto backup report.

8.4.2 autoRestoreReport

The autoRestoreReport notification is emitted from the managed object when an automatic restore is occurred on the information represented by this object instance. The criteria that triggers the automatic restore is system specific. The source of the restored information and the result of the restore operation (i.e. either success or failure) shall be reported in the notification.

This service uses the M-EVENT-REPORT service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Event Type parameter shall indicate auto restore report.

8.4.3 deliverResultNotification

This notification type is used to report deliver results and is emitted from the managed object when the delivery is completed. The deliver result parameter may take one of the following values: pass, fail or unknown.

This service uses the M-EVENT-REPORT service and procedures defined in CCITT Rec. X.710 and ISO/IEC 9595. The Event Type parameter shall indicate deliver result.

8.5 Parameters

8.5.1 alarmEffectOnServiceParameter

The alarmEffectOnServiceParameter is a parameter to be included in the ManagementExtension parameter of the AdditionalInformation parameter of the AlarmInfo parameter in an Alarm Reporting service. This parameter indicates whether the condition which caused the alarm will affect service.

The ManagementExtension is of the form (see CCITT Rec. X.721 | ISO/IEC 10165-2):

```
ManagementExension ::= SEQUENCE {
identifier OBJECT IDENTIFIER,
significance [1] BOOLEAN DEFAULT FALSE,
information [2] ANY DEFINED BY identifier}
```

The OBJECT IDENTIFIER carried in identifier shall be the value under which this parameter definition is registered. The type carried in information shall be the type identified by the WITH SYNTAX construct of this parameter definition.

8.5.2 softwareProcessingFailureParameter

The softwareProcessingFailureParameter defines the data syntax to be return in a CMIP ProcessingFailure error reply to a M-ACTION if a request for an operation on software is denied due to errors other than those already defined in CMIP-1. The attributes in the returned data syntax include the state attributes of the emitting object. Any other applicable attributes may be included but these are a local matter.

8.6 Name bindings

8.6.1 Software distributor – Subsystem

This name binding is used for naming a software distributor object with respect to a subsystem object.

8.6.2 Software distributor – System

This name binding is used for naming a software distributor object with respect to a system object.

8.6.3 Software unit – Subsystem

This name binding is used for naming a software unit object with respect to a subsystem object.

8.6.4 Software unit – System

This name binding is used for naming a software unit object with respect to a system object.

9 Service definitions

9.1 Introduction

This clause defines services to deliver software, execute program, install software, revert software, terminate validation of software and validate software. It also defines services reporting automatic software backups and restoration. In addition, it defines generic services for backup and restore.

9.2 Backup service

The Backup service allows a manager to request that another open system (the managed system) backup an entity. Table 6 lists the parameters for this service.

The Backup service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

The Backup reply parameter shall be present in a positive response, otherwise the Errors parameter shall be present.

9.3 Deliver service

The Deliver service allows a manager to request that another open system (the managed system) initiate software delivery. Table 7 lists the parameters for this service.

The Deliver service uses the parameters defined in clause 8 of this Recommendation | International Standard in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

The Deliver result info parameter shall be present in a positive response, otherwise the Errors parameter shall be present.

Table 6 – Backup parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Backup type	M	C(=)
Backup argument	M	–
Backup destination	M	–
Additional information	U	–
Current time	–	P
Backup reply	–	C
Reply	–	C
Additional information	–	U
Errors	–	C

Table 7 – Deliver parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Deliver type	M	C(=)
Deliver info	M	–
Deliver id	U	–
Target software	M	–
Target system	U	–
Transfer info	U	–
Additional info	U	–
Current time	–	P
Deliver result info	–	C
Deliver id	–	U
Deliver result	–	C
Additional info	–	U
Errors	–	C

9.4 Execute program service

The Execute program service allows a manager to request that another open system (the managed system) execute program software. Table 8 lists the parameters for this service.

The Execute program service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

The Execute program reply parameter shall be present in a positive response, otherwise the Errors parameter shall be present.

Table 8 – Execute program parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Execute program type	M	C(=)
Execute program info	M	–
Current time	–	P
Execute program reply	–	C
Process id	–	C
Process owner	–	C
Start time	–	C
Additional info	–	U
Errors	–	C

9.5 Install service

The Install service allows a manager to request that another open system (the managed system) install software. Table 9 lists the parameters for this service.

The Install service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

If there is not a positive response, the Errors parameter shall be present in the response.

9.6 Restore service

The Restore service allows a manager to request that another open system (the managed system) restore an entity. Table 10 lists the parameters for this service.

The Restore service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

If there is not a positive response, the Errors parameter shall be present in the response.

Table 9 – Install parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Install type	M	C(=)
Install Info	M	–
Target software	M	–
Install info	M	–
Current time	–	P
Errors	–	C

Table 10 – Restore parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Restore type	M	C(=)
Restore argument	M	–
Restore source	M	–
Additional info	U	–
Current time	–	P
Errors	–	C

9.7 Revert service

The Revert service allows a manager to request that another open system (the managed system) revert software. Table 11 lists the parameters for this service.

The Revert service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

The Revert reply parameter shall be present in a positive response, otherwise the Errors parameter shall be present.

Table 11 – Revert parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Revert type	M	C(=)
Revert info	M	–
Patch id	M	–
Additional info	U	–
Current time	–	P
Revert reply	–	C
Applied patches	–	C
Additional info	–	U
Errors	–	C

9.8 Terminate validation service

The Terminate validation service allows a manager to request that another open system (the managed system) terminate software validation. Table 12 lists the parameters for this service.

The Terminate validation service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

The Terminate validation reply parameter shall be present in a positive response, otherwise the Errors parameter shall be present.

9.9 Validate service

The Validate service allows a manager to request that another open system (the managed system) validate software. Table 13 lists the parameters for this service.

The Validate service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

The Validate reply parameter shall be present in a positive response, otherwise the Errors parameter shall be present.

9.10 Auto backup report service

The Auto backup report service allows one open system (the managed system) to report automatic backup. Table 14 lists the parameters for this service.

The Auto backup report service uses the parameters defined in clause 8 in addition to the general M-EVENT-REPORT service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

9.11 Auto restore report service

The Auto restore report service allows one open system (the managed system) to report automatic restore. Table 15 lists the parameters for this service.

The Auto restore report service uses the parameters defined in clause 8 in addition to the general M-EVENT-REPORT service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

Table 12 – Terminate validation parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Terminate validation type	M	C(=)
Terminate validation argument	M	–
Terminate validation info	M	–
Additional info	U	–
Current time	–	P
Terminate validation reply	–	C
Errors	–	C

Table 13 – Validate parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Validate type	M	C(=)
Validate info	M	–
Current time	–	P
Validate reply	–	C
Errors	–	C

Table 14 – Auto backup report parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Mode	P	–
Managed object class	–	P
Managed object instance	–	P
Auto backup report type	M	C(=)
Event time	P	–
Auto backup report info	M	–
Backup result	M	–
Additional info	U	–
Current time	–	P
Event reply	–	–
Errors	–	C

Table 15 – Auto restore report parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Mode	P	–
Managed object class	–	P
Managed object instance	–	P
Auto restore report type	M	C(=)
Event time	P	–
Auto restore report info	M	–
Source	M	–
Success	M	–
Additional info	U	–
Current time	–	P
Event reply	–	–
Errors	–	C

9.12 Deliver result notification service

The Deliver result notification service allows one open system (the managed system) to report the result of delivery of software. Table 16 lists the parameters for this service.

The Deliver result notification service uses the parameters defined in clause 8 in addition to the general M-EVENT-REPORT service parameters defined in CCITT Rec. X.710 and ISO/IEC 9595.

Table 16 – Deliver result notification parameters

Parameter Name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Mode	P	–
Managed object class	–	P
Managed object instance	–	P
Deliver result notification type	M	C(=)
Event time	P	–
Deliver result info	M	–
Deliver id	U	–
Deliver result	M	–
Additional info	U	–
Current time	–	P
Event reply	–	–
Errors	–	C

10 Functional units

Three functional units are defined in this Recommendation | International Standard for the management of software and other entities:

a) software control functional unit;

b) software deliver functional unit;

c) backup restore functional unit.

The software control functional unit requires support of the execute program, install, revert, terminate validation, and validation services. The software deliver functional unit requires support of the deliver and deliver result notification services. The backup restore functional unit requires support of the backup action, restore action, auto backup report notification, and auto restore report notification services.

11 Protocol

11.1 Elements of procedure

11.1.1 Backup procedure

11.1.1.1 Manager role

11.1.1.1.1 Invocation

The backup procedures are initiated by the backup primitive. On receipt of a backup primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the backup primitive. The confirmed mode shall be used.

11.1.1.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a backup operation, the SMAPM shall issue a backup confirmation primitive to the backup service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the backup procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The backup service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.1.2 Agent role

11.1.1.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the backup service, the SMAPM shall, if the MAPDU is well formed, issue a backup indication primitive to the backup service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.1.2.2 Response

The SMAPM shall accept a backup response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the backup response primitive.

11.1.2 Deliver procedure

11.1.2.1 Manager role

11.1.2.1.1 Invocation

The deliver procedures are initiated by the deliver primitive. On receipt of a deliver primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the deliver primitive. The confirmed mode shall be used.

11.1.2.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a deliver operation, the SMAPM shall issue a deliver confirmation primitive to the deliver service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the deliver procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The deliver service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.2.2 Agent role

11.1.2.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the deliver service, the SMAPM shall, if the MAPDU is well formed, issue a deliver indication primitive to the deliver service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.2.2.2 Response

The SMAPM shall accept a deliver response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the deliver response primitive.

11.1.3 Execute program procedure

11.1.3.1 Manager role

11.1.3.1.1 Invocation

The execute program procedures are initiated by the execute program primitive. On receipt of an execute program primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the execute program primitive. The confirmed mode shall be used.

11.1.3.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to an execute program operation, the SMAPM shall issue an execute program confirmation primitive to the execute program service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the execute program procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The execute program service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.3.2 Agent role

11.1.3.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the execute program service, the SMAPM shall, if the MAPDU is well formed, issue an execute program indication primitive to the execute program service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.3.2.2 Response

The SMAPM shall accept an execute program response primitive and shall construct an MAPDU confirming the operation and issue CMIS M-ACTION response service primitive with parameters derived from the execute program response primitive.

11.1.4 Install procedure

11.1.4.1 Manager role

11.1.4.1.1 Invocation

The install procedures are initiated by the install primitive. On receipt of an install primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the install primitive. The confirmed mode shall be used.

11.1.4.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to an install operation, the SMAPM shall issue an install confirmation primitive to the install service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the install procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The install service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.4.2 Agent role

11.1.4.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the install service, the SMAPM shall, if the MAPDU is well formed, issue an install indication primitive to the install service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.4.2.2 Response

The SMAPM shall accept an install response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the install response primitive.

11.1.5 Restore procedure

11.1.5.1 Manager role

11.1.5.1.1 Invocation

The restore procedures are initiated by the restore primitive. On receipt of a restore primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the restore primitive. The confirmed mode shall be used.

11.1.5.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a restore operation, the SMAPM shall issue a restore confirmation primitive to the restore service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the restore procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The restore service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.5.2 Agent role

11.1.5.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the restore service, the SMAPM shall, if the MAPDU is well formed, issue a restore indication primitive to the restore service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.5.2.2 Response

The SMAPM shall accept a restore response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the restore response primitive.

11.1.6 Revert procedure

11.1.6.1 Manager role

11.1.6.1.1 Invocation

The revert procedures are initiated by the revert primitive. On receipt of a revert primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the revert primitive. The confirmed mode shall be used.

11.1.6.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a revert operation, the SMAPM shall issue a revert confirmation primitive to the revert service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the revert procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The revert service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.6.2 Agent role

11.1.6.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the revert service, the SMAPM shall, if the MAPDU is well formed, issue a revert indication primitive to the revert service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.6.2.2 Response

The SMAPM shall accept a revert response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the revert response primitive.

11.1.7 Terminate validation procedure

11.1.7.1 Manager role

11.1.7.1.1 Invocation

The terminate validation procedures are initiated by the terminate validation primitive. On receipt of a terminate validation primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the terminate validation primitive. The confirmed mode shall be used.

11.1.7.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a terminate validation operation, the SMAPM shall issue a terminate validation confirmation primitive to the terminate validation service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the terminate validation procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The terminate validation service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.7.2 Agent role

11.1.7.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the terminate validation service, the SMAPM shall, if the MAPDU is well formed, issue a terminate validation indication primitive to the terminate validation service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.7.2.2 Response

The SMAPM shall accept a terminate validation response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the terminate validation response primitive.

11.1.8 Validate procedure

11.1.8.1 Manager role

11.1.8.1.1 Invocation

The validate procedures are initiated by the validate primitive. On receipt of a validate primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the validate primitive. The confirmed mode shall be used.

11.1.8.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a validate operation, the SMAPM shall issue a validate confirmation primitive to the validate service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the validate procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The validate service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.8.2 Agent role

11.1.8.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the validate service, the SMAPM shall, if the MAPDU is well formed, issue a validate indication primitive to the validate service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.8.2.2 Response

The SMAPM shall accept a validate response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the validate response primitive.

11.1.9 Auto backup report procedures

11.1.9.1 Agent role

11.1.9.1.1 Invocation

The auto backup report procedures are initiated by the auto backup report request primitive. On receipt of an auto backup report request primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-EVENT-REPORT request service primitive with parameters derived from the auto backup report request primitive. In the non-confirmed mode, the procedure in 11.1.9.1.2 does not apply.

11.1.9.1.2 Receipt of response

On receipt of a CMIS M-EVENT-REPORT confirm service primitive containing an MAPDU responding to an auto backup report notification, the SMAPM shall issue an auto backup report confirmation primitive to the auto backup report service user with parameters derived from the CMIS M-EVENT-REPORT confirm service primitive, thus completing the auto backup report procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The auto backup report service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.9.2 Manager role

11.1.9.2.1 Receipt of request

On receipt of a CMIS M-EVENT-REPORT indication service primitive containing an MAPDU requesting the auto backup report service, the SMAPM shall, if the MAPDU is well formed, issue an auto backup report indication primitive to the auto backup report service user with parameters derived from the CMIS M-EVENT-REPORT indication service primitive. Otherwise, the SMAPM shall, in the confirmed mode, construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-EVENT-REPORT response service primitive with an error parameter present. In the non-confirmed mode, the procedure in 11.1.9.2.2 does not apply.

11.1.9.2.2 Response

In the confirmed mode, the SMAPM shall accept an auto backup report response primitive and shall construct an MAPDU confirming the notification and issue a CMIS M-EVENT-REPORT response service primitive with parameters derived from the auto backup report response primitive.

11.1.10 Auto restore report procedures

11.1.10.1 Agent role

11.1.10.1.1 Invocation

The auto restore report procedures are initiated by the auto restore report request primitive. On receipt of an auto restore report request primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-EVENT-REPORT request service primitive with parameters derived from the auto restore report request primitive. In the non-confirmed mode, the procedure in 11.1.10.1.2 does not apply.

11.1.10.1.2 Receipt of response

On receipt of a CMIS M-EVENT-REPORT confirm service primitive containing an MAPDU responding to an auto restore report notification, the SMAPM shall issue an auto restore report confirmation primitive to the auto restore report service user with parameters derived from the CMIS M-EVENT-REPORT confirm service primitive, thus completing the auto restore report procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The auto restore report service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.10.2 Manager role

11.1.10.2.1 Receipt of request

On receipt of a CMIS M-EVENT-REPORT indication service primitive containing an MAPDU requesting the auto restore report service, the SMAPM shall, if the MAPDU is well formed, issue an auto restore report indication primitive to the auto restore report service user with parameters derived from the CMIS M-EVENT-REPORT indication service primitive. Otherwise, the SMAPM shall, in the confirmed mode, construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-EVENT-REPORT response service primitive with an error parameter present. In the non-confirmed mode, the procedure in 11.1.10.2.2 does not apply.

11.1.10.2.2 Response

In the confirmed mode, the SMAPM shall accept an auto restore report response primitive and shall construct an MAPDU confirming the notification and issue a CMIS M-EVENT-REPORT response service primitive with parameters derived from the auto restore report response primitive.

11.1.11 Deliver result notification procedures

11.1.11.1 Agent role

11.1.11.1.1 Invocation

The deliver result notification procedures are initiated by the deliver result notification request primitive. On receipt of a deliver result notification request primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-EVENT-REPORT request service primitive with parameters derived from the deliver result notification request primitive. In the non-confirmed mode, the procedure in 11.1.11.1.2 does not apply.

11.1.11.1.2 Receipt of response

On receipt of a CMIS M-EVENT-REPORT confirm service primitive containing an MAPDU responding to a deliver result notification, the SMAPM shall issue a deliver result notification confirmation primitive to the deliver result notification service user with parameters derived from the CMIS M-EVENT-REPORT confirm service primitive, thus completing the deliver result notification procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The deliver result notification service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.11.2 Manager role

11.1.11.2.1 Receipt of request

On receipt of a CMIS M-EVENT-REPORT indication service primitive containing an MAPDU requesting the deliver result notification service, the SMAPM shall, if the MAPDU is well formed, issue a deliver result notification indication primitive to the deliver result notification service user with parameters derived from the CMIS M-EVENT-REPORT indication service primitive. Otherwise, the SMAPM shall, in the confirmed mode, construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-EVENT-REPORT response service primitive with an error parameter present. In the non-confirmed mode, the procedure in 11.1.11.2.2 does not apply.

11.1.11.2.2 Response

In the confirmed mode, the SMAPM shall accept a deliver result notification response primitive and shall construct an MAPDU confirming the notification and issue a CMIS M-EVENT-REPORT response service primitive with parameters derived from the deliver result notification response primitive.

11.2 Abstract syntax

11.2.1 Objects

This Recommendation | International Standard references the following support objects whose ASN.1 value notation is specified in Annex A:

a) softwareUnit;
b) executableSoftware;
c) softwareDistributor.

11.2.2 Packages

This Recommendation | International Standard references the following package definitions whose ASN.1 value notation is specified in Annex A:

a) appliedPatchPackage;
b) checkSumPackage;
c) executeProgramPackage;
d) fileInformationPackage;
e) filePackage;
f) informationAutoBackupPackage;
g) informationAutoRestorePackage;
h) informationBackupPackage;
i) informationRestorePackage;
j) installPackage;
k) noteFieldPackage;
l) processingErrorAlarmOnServicePackage;
m) revertPackage;
n) terminateValidationPackage;
o) usageStatePackage;
p) validationPackage.

11.2.3 Attributes

This Recommendation | International Standard references the following specific management attributes, the abstract syntax for which is specified in Annex A:

a) appliedPatches;

b) checkSum;

c) dateDelivered;

d) dateInstalled;

e) dateOfCreation;

f) dateOfLastModification;

g) fileLocation;

h) fileSize;

i) fileType;

j) futureAutoBackupDestination;

k) futureAutoBackupTriggerThreshold;

l) futureAutoRestoreAllowed;

m) futureAutoRestoreSource;

n) identityOfCreator;

o) identityOfLastModifier;

p) lastBackupDestination;

q) lastBackupTime;

r) lastRestoreSource;

s) lastRestoreTime;

t) noteField;

u) softwareDistributorId.

11.2.4 Notifications

This Recommendation | International Standard references the following specific notification types, the abstract syntax for which is specified in Annex A:

a) autoBackupReport;

b) autoRestoreReport;

c) deliverResultNotification.

11.2.5 Actions

This Recommendation | International Standard references the following specific action types, the abstract syntax for which is specified in Annex A:

a) backup;

b) deliver;

c) executeProgram;

d) install

e) restore;

f) revert:

g) terminateValidation;

h) validate.

11.2.6 Name bindings

This Recommendation | International Standard references the following specific name bindings, the abstract syntax for which is specified in Annex A:

a) softwareDistributor-subsystem;

b) softwareDistributor-system;

c) softwareUnit-subsystem;

d) softwareUnit-system.

11.3 Negotiation of functional units

This Recommendation | International Standard assigns the following object identifier:

{joint-iso-ccitt ms(9) function(2) part18(18) functionalUnitPackage(1)}

as a value of the ASN.1 type FunctionalUnitPackageId defined in CCITT Rec. X.701 | ISO/IEC 10040 for negotiating the following functional units:

0 software control functional unit

1 software deliver functional unit

2 backup restore functional unit

where the number identifies the bit position assigned to the functional unit, and the name references the functional unit as defined in clause 10.

Within the Systems management application context, the mechanism for negotiating the alarm reporting functional unit is described by CCITT Rec. X.701 | ISO/IEC 10040.

NOTE – The requirement to negotiate functional units is specified by the application context.

12 Relationship with other functions

The following functions are provided by other Systems Management Functions:

- performance of software, covered by the Summarization Function (see ITU-T Rec. X.738 | ISO/IEC 10164-13) and the Metric Objects and Attributes Function (see ITU-T Rec. X.739 | ISO/IEC 10164-11);
- software audit trailing, covered by the Security Audit Trail Function (see CCITT Rec. X.740 | ISO/IEC 10164-8);
- support for software security, covered by Objects and Attributes for Access Control (see ITU-T Rec. X.741 | ISO/IEC 10164-9);
- accounting of software usage, covered by the Usage Metering Function for Accounting Purposes (see ITU-T Rec. X.742 | ISO/IEC 10164-10);
- Software Testing (includes Installation of the test environment, Test run of the software, Setting breakpoints, and suspending and resuming the software test environment), covered by the Test Management Function (see ITU-T Rec. X.745 | ISO/IEC 10164-12);
- scheduling of software functions and operations, covered by the Scheduling Function (see ITU-T Rec. X.746 | ISO/IEC 10164-15).

13 Conformance

Implementations claiming to conform to this Recommendation | International Standard shall comply with the conformance requirements as defined in the following subclauses.

13.1 Static conformance

The implementation shall conform to the requirements of this Recommendation | International Standard in the manager role, the agent role, or both roles. A claim of conformance to at least one role shall be made in Table B.1.

If a claim of conformance is made for support in the manager role, the implementation shall support at least one management operation on, or notification from, any of the managed objects or subclasses of managed objects defined in this Recommendation | International Standard. The conformance requirements in the manager role for those management operations and notifications are specified in Annex B and further tables referenced by Annex B.

If a claim of conformance is made for support in the agent role, the implementation shall support one or more instances of the managed object classes or subclasses thereof specified in Table B.4. In the agent role, conformance may also be claimed to subclasses of the log record object.

The implementation shall support the transfer syntax derived from the encoding rules specified in CCITT Rec. X.209 and ISO/IEC 8825 named {joint-iso-ccitt asn1(1) basicEncoding(1)} for the abstract data types referenced by the definitions for which support is claimed.

13.2 Dynamic conformance

Implementations claiming to conform to this Recommendation | International Standard shall support the elements of procedure and definitions of semantics corresponding to the definitions for which support is claimed.

13.3 Management implementation conformance statement requirements

Any MCS proforma, MICS proforma, MOCS proforma, and MRCS proforma which conforms to this Recommendation | International Standard shall be technically identical to the proformas specified in Annexes B, C, D and E preserving table numbering and the index numbers of items, and differing only in pagination and page headers.

The supplier of an implementation which is claimed to conform to this Recommendation | International Standard shall complete a copy of the Management Conformance Summary (MCS) provided in Annex B as part of the conformance requirements together with any other ICS proformas referenced as applicable from that MCS. An ICS which conforms to this Recommendation | International Standard shall:

- describe an implementation which conforms to this Recommendation | International Standard;
- have been completed in accordance with the instructions for completion given in ITU-T Rec. X.724 | ISO/IEC 10165-6;
- include the information necessary to uniquely identify both the supplier and the implementation.

Claims of conformance to the management information defined in this Recommendation | International Standard in managed object classes defined elsewhere shall include the requirements of the MIDS proforma, as specified in Annex F, in the MOCS for the managed object class.

Annex A

Definition of management information

(This annex forms an integral part of this Recommendation | International Standard)

--A.1 Managed Object Classes

--A.1.1 Software

-- This object class is defined in Rec. M.3100.

-- (See Annex C.)

--A.1.2 softwareUnit

```
softwareUnit          MANAGED OBJECT CLASS

        DERIVED FROM "Rec. M.3100:1995":software;

        CHARACTERIZED BY
            "Rec. M.3100:1995":administrativeOperationalStatesPackage,
            softwareUnitPackage     PACKAGE
                BEHAVIOUR softwareUnitBehaviour;
                ATTRIBUTES
                "Rec. X.721 | ISO/IEC 10165-2:1992":availabilityStatus  GET,
                "Rec. X.721 | ISO/IEC 10165-2:1992":proceduralStatus  GET;;
            processingErrorAlarmOnServicePackage;

        CONDITIONAL PACKAGES
            appliedPatchPackage PRESENT IF "an instance supports software patching",
            checkSumPackage PRESENT IF "an instance supports check sum validation",
            fileInformationPackage PRESENT IF "an instance supports file information",
            filePackage PRESENT IF "an instance supports representation of a file",
            informationAutoBackupPackage PRESENT IF "an instance supports automatic backup",
            informationAutoRestorePackage PRESENT IF "an instance supports automatic restore",
            informationBackupPackage PRESENT IF "an instance supports the backup operation",
            informationRestorePackage PRESENT IF "an instance supports the restore operation",
            installPackage PRESENT IF "an instance supports the install operation",
            noteFieldPackage PRESENT IF "an instance supports it",
            revertPackage PRESENT IF "an instance supports it",
            terminateValidationPackage PRESENT IF "the validationPackage is present and an instance supports it",
            usageStatePackage PRESENT IF "an instance supports it",
            validationPackage PRESENT IF "an instance supports it";

REGISTERED AS    {SWMF.softwareManagement  managedObjectClass(3) softwareUnit(1)};

        softwareUnitBehaviour BEHAVIOUR

            DEFINED AS
```

"The softwareUnit object class is a class of managed objects that provide administrable information associated with software (whether it be in the form of an executable file, such as program software, or a non-executable file, such as a data or cross-connect mapping table). The file type, file location, and file size are among the attributes identified in this object class. When the fileInformationPackage is present, the mandatory initial value of the dateOfCreation attribute is the time that the managed object is created.

When the attribute value change notification package (inherited from the superclass software) is present, the attributeValueChange notification defined in Recommendation X.721 shall be emitted when the value of one of the following attribute changes:

- futureAutoBackupTriggerThreshold;
- futureAutoBackupDestination;
- futureAutoRestoreSource; and
- futureAutoRestoreAllowed.

Because some of the above attributes are in conditional packages, the behaviour for emitting the attributeValueChange notification applies only when the corresponding conditional packages are present in the managed object.";

--A.1.3 executableSoftware

```
executableSoftware        MANAGED OBJECT CLASS

        DERIVED FROM softwareUnit;

        CHARACTERIZED BY
        executableSoftwarePackage        PACKAGE
            BEHAVIOUR executableSoftwareBehaviour;;,
            usageStatePackage;

        CONDITIONAL        PACKAGES
            executeProgramPackage
                PRESENT IF "an instance supports it";

REGISTERED AS    { SWMF.softwareManagement  managedObjectClass(3)  executableSoftware(2)};

executableSoftwareBehaviour BEHAVIOUR

        DEFINED AS
```

"The executableSoftware object class is a class of managed objects that provide administrable information associated with an executable program in the managed system. The actual executable program (that may consist of code segments with or without data segments, etc.) may be in a non-standard, machine-dependent format that is generally unreadable by managing system and the rest of the outside world. An action called executeProgram (conditionally) can be used to execute the program represented by the executableSoftware object instance. The usageState attribute is used to indicate if there are any active executions of the program.";

--A.1.4 softwareDistributor

```
softwareDistributor        MANAGED OBJECT CLASS

DERIVED FROM     "Rec. X.721 | ISO/IEC 10165-2:1992":top;

        CHARACTERIZED BY
            softwareDistributorPackage    PACKAGE
                BEHAVIOUR softwareDistributorBehaviour;
                ATTRIBUTES
                    "Rec. X.721 | ISO/IEC 10165-2:1992":administrativeState GET-REPLACE,
                    "Rec. X.721 | ISO/IEC 10165-2:1992":operationalState GET,
                    softwareDistributorId GET;
                ACTIONS
                    deliver;
                NOTIFICATIONS
                    deliverResultNotification,
                    "Rec. X.721 | ISO/IEC 10165-2:1992":objectCreation,
                    "Rec. X.721 | ISO/IEC 10165-2:1992":objectDeletion,
                    "Rec. X.721 | ISO/IEC 10165-2:1992":stateChange ;;;

REGISTERED AS    {SWMF.softwareManagement  managedObjectClass(3)  softwareDistributor(3)};

softwareDistributorBehaviour BEHAVIOUR

        DEFINED AS
```

"A Software distributor managed object is a managed object which distributes software to the target managed system when it receives a deliver operation from the managing system. This managed object notifies the result of the distribution when the distribution is terminated. The stateChangeNotification defined in Rec. X.721 | ISO/IEC 10165 2:1992 shall be emitted if the value of the administrative state or operational state changes. The Software distributor managed object class is a subclass of the top managed object class defined in Rec. X.721 | ISO/IEC 10165-2:1992.";

--A.2 Packages

--A.2.1 appliedPatchPackage

```
appliedPatchPackage     PACKAGE

        ATTRIBUTES
            appliedPatches    GET;

REGISTERED AS    {SWMF.softwareManagement  package(4)  appliedPatchPackage(1)};
```

```
--A.2.2  checkSumPackage

checkSumPackage        PACKAGE

        ATTRIBUTES
            checkSum
                GET;

REGISTERED AS   {SWMF.softwareManagement package(4) checkSumPackage(2)};

--A.2.3  executeProgramPackage

executeProgramPackage       PACKAGE

        ACTIONS
            executeProgram;

REGISTERED AS   {SWMF.softwareManagement package(4) executeProgramPackage(3)};

--A.2.4  fileInformationPackage

fileInformationPackage        PACKAGE

        ATTRIBUTES
            dateOfCreation  GET,
            identityOfCreator  GET,
            dateOfLastModification  GET,
            identityOfLastModifier  GET,
            dateDelivered  GET,
            dateInstalled  GET;

REGISTERED AS   {SWMF.softwareManagement package(4) fileInformationPackage(4)};

--A.2.5  filePackage

filePackage    PACKAGE

        ATTRIBUTES
            fileLocation  GET,
            fileSize  GET,
            fileType  GET;

REGISTERED AS   {SWMF.softwareManagement package(4) filePackage(5)};

--A.2.6  informationAutoBackupPackage

informationAutoBackupPackage    PACKAGE

        ATTRIBUTES
            futureAutoBackupTriggerThreshold        GET-REPLACE,
            futureAutoBackupDestination             GET-REPLACE;

        NOTIFICATIONS
            autoBackupReport;

REGISTERED AS   {SWMF.softwareManagement package(4) informationAutoBackupPackage(6)};

--A.2.7  informationAutoRestorePackage

informationAutoRestorePackage     PACKAGE

        ATTRIBUTES
            futureAutoRestoreSource                 GET-REPLACE,
            futureAutoRestoreAllowed                GET-REPLACE;

        NOTIFICATIONS
            autoRestoreReport;

REGISTERED AS   {SWMF.softwareManagement package(4) informationAutoRestorePackage(7)};
```

```
--A.2.8  informationBackupPackage

informationBackupPackage      PACKAGE

        ATTRIBUTES
            lastBackupTime          GET,
            lastBackupDestination   GET;

        ACTIONS
            backup softwareProcessingFailureParameter;

REGISTERED AS   {SWMF.softwareManagement package(4) informationBackupPackage(7)};

--A.2.9  informationRestorePackage

informationRestorePackage      PACKAGE

        ATTRIBUTES
            lastRestoreTime         GET,
            lastRestoreSource       GET;

        ACTIONS
            restore;

REGISTERED AS   {SWMF.softwareManagement package(4) informationRestorePackage(9)};

--A.2.10  installPackage

installPackage      PACKAGE

        ACTIONS
            install;

REGISTERED AS   {SWMF.softwareManagement package(4) installPackage(10)};

--A.2.11  noteFieldPackage

noteFieldPackage      PACKAGE

        ATTRIBUTES
            noteField     GET-REPLACE;

REGISTERED AS   {SWMF.softwareManagement package(4) noteFieldPackage(11)};

--A.2.12  processingErrorAlarmOnServicePackage

processingErrorAlarmOnServicePackage      PACKAGE

        NOTIFICATIONS
            "Rec. X.721 | ISO/IEC 10165-2:1992":processingErrorAlarm
                alarmEffectOnServiceParameter;

REGISTERED AS   {SWMF.softwareManagement package(4) processingErrorAlarmOnServicePackage(12)};

--A.2.13  revertPackage

revertPackage      PACKAGE

        ACTIONS
            revert softwareProcessingFailureParameter ;

REGISTERED AS   {SWMF.softwareManagement package(4) revertPackage(13)};

--A.2.14  terminateValidationPackage

terminateValidationPackage      PACKAGE

        ACTIONS
            terminateValidation;

REGISTERED AS   {SWMF.softwareManagement package(4) terminateValidationPackage(14)};
```

--A.2.15 usageStatePackage

```
usageStatePackage        PACKAGE
        ATTRIBUTES
            "Rec. X.721 | ISO/IEC 10165-2:1992":usageState    GET;
REGISTERED AS    {SWMF.softwareManagement  package(4)  usageStatePackage(15)};
```

--A.2.16 validationPackage

```
validationPackage        PACKAGE
        ACTIONS
            validate;
REGISTERED AS    {SWMF.softwareManagement  package(4)  validatePackage(16)};
```

--A.3 Attributes

--A.3.1 appliedPatches

```
appliedPatches      ATTRIBUTE
        WITH ATTRIBUTE SYNTAX SWMF.AppliedPatches;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR appliedPatchesBehaviour;
REGISTERED AS    {SWMF.softwareManagement  attribute(7)  appliedPatches(1)};
appliedPatchesBehaviour BEHAVIOUR
        DEFINED AS
"This attribute identifies the patches that have been applied to and still exist in the software which is represented by the software unit object instance. Patches are updates to executable programs. The value of this attribute is read-only and automatically updated when a patch is applied to the software.";
```

--A.3.2 checkSum

```
checkSum            ATTRIBUTE
        WITH ATTRIBUTE SYNTAX SWMF.CheckSum;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR checkSumBehaviour;
REGISTERED AS    {SWMF.softwareManagement  attribute(7)  checkSum(2)};
checkSumBehaviour BEHAVIOUR
        DEFINED AS
"This attribute identifies the checksum of the software information represented by the softwareUnit object instance.";
```

--A.3.3 dateDelivered

```
dateDelivered       ATTRIBUTE
WITH ATTRIBUTE SYNTAX SWMF.Date;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR dateDeliveredBehaviour;
REGISTERED AS    {SWMF.softwareManagement  attribute(7)  dateDelivered(3)};
dateDeliveredBehaviour BEHAVIOUR
        DEFINED AS
"This attribute identifies the time that the information represented by the softwareUnit object instance was delivered to the managed system. Valid values for this attribute are ASN.1 GeneralizedTime or NULL if the information has not been delivered.";
```

--A.3.4 dateInstalled

dateInstalled ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.Date;

MATCHES FOR EQUALITY;

BEHAVIOUR dateInstalledBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) dateInstalled(4)};

dateInstalledBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the time that the information represented by the softwareUnit object instance was installed. Valid values for this attribute are ASN.1 GeneralizedTime or NULL if the information has not been installed.";

--A.3.5 dateOfCreation

dateOfCreation ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.GlobalTime;

MATCHES FOR EQUALITY;

BEHAVIOUR dateOfCreationBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) dateOfCreation(5)};

dateOfCreationBehaviour BEHAVIOUR

DEFINED AS

"This attribute indicates the time of creation of the managed object. The syntax of this attribute is of ASN.1 GeneralizedTime type.";

--A.3.6 dateOfLastModification

dateOfLastModification ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.Date;

MATCHES FOR EQUALITY;

BEHAVIOUR dateOfLastModificationBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) dateOfLastModification(6)};

dateOfLastModificationBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the time of the last, or most recent modification to the information represented by the softwareUnit object instance. Valid values for this attribute are ASN.1 GeneralizedTime or NULL if the information has not been modified.";

--A.3.7 fileLocation

fileLocation ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.FileLocation;

MATCHES FOR EQUALITY;

BEHAVIOUR fileLocationBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) fileLocation(7)};

fileLocationBehaviour BEHAVIOUR

DEFINED AS

"This attribute specifies the full address(es) (either logical or physical) of the softwareUnit object. The format of the address is implementation-dependent, conforming to the file-addressing conventions of the particular managed system in question. A null value of this attribute indicates that the information to which the softwareUnit managed object applies has not yet been installed in the managed system.";

--A.3.8 fileSize

fileSize ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.InformationSize;

MATCHES FOR EQUALITY;

BEHAVIOUR fileSizeBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) fileSize(8)};

fileSizeBehaviour BEHAVIOUR

DEFINED AS

"This attribute indicates the size of the softwareUnit managed object.";

--A.3.9 fileType

fileType ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.FileType;

MATCHES FOR EQUALITY;

BEHAVIOUR fileTypeBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) fileType(9)};

fileTypeBehaviour BEHAVIOUR

DEFINED AS

"This attribute indicates the type of the softwareUnit. Possible softwareUnit types are unstructured binary file (e.g. executable file), unstructured text file (e.g. non-executable file), and block special file, etc.";

--A.3.10 futureAutoBackupDestination

futureAutoBackupDestination ATTRIBUTE

WITH ATTRIBUTE SYNTAX BackupRestoreASN1Module.BackupDestination;

MATCHES FOR EQUALITY;

BEHAVIOUR futureAutoBackupDestinationBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) futureAutoBackupDestination(10)};

futureAutoBackupDestinationBehaviour BEHAVIOUR

DEFINED AS

"This attribute specifies the destination to which the information represented by this object instance will be backup. The backup criteria is defined in the futureAutoBackupTriggerThreshold attribute of the object instance. The destination can be another object instance of the same object class exists in the same local managed system, a remote open system (by using a particular file transfer protocol, e.g. FTAM), or the managing system (by in-line using the autoBackupReport notification).";

--A.3.11 futureAutoBackupTriggerThreshold

futureAutoBackupTriggerThreshold ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.Integer;

MATCHES FOR EQUALITY, ORDERING;

BEHAVIOUR futureAutoBackupTriggerThresholdBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) futureAutoBackupTriggerThreshold(11)};

futureAutoBackupTriggerThresholdBehaviour BEHAVIOUR

DEFINED AS

"This attribute specifies the threshold that will trigger an automatic backup for the information represented by the object instance. The threshold is defined as the number of time that the information has been modified. Once the information has been modified for that number of times, an automatic backup will be performed. The backup destination is specified in the attribute futureAutomacticBackupDestination. Such backups are carried out in addition to other scheduled periodic backup. At the completion of the automatic backup, an autoBackupReport notification shall be emitted from the object.";

--A.3.12 futureAutoRestoreAllowed

```
futureAutoRestoreAllowed        ATTRIBUTE
        WITH ATTRIBUTE SYNTAX SWMF.Boolean;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR futureAutoRestoreAllowedBehaviour;
REGISTERED AS    {SWMF.softwareManagement  attribute(7)  futureAutoRestoreAllowed(12)};
futureAutoRestoreAllowedBehaviour BEHAVIOUR
        DEFINED AS
```

"This attribute specifies whether automatic restore of the information represented by this manage object instance is allowed. The syntax of this attribute is of ASN.1 BOOLEAN type with the value TRUE meaning allowed, and FALSE meaning not allowed. The criteria that triggers automatic information restore is system specific.";

--A.3.13 futureAutoRestoreSource

```
futureAutoRestoreSource         ATTRIBUTE
        WITH ATTRIBUTE SYNTAX SWMF.AutoRestoreSource;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR futureAutoRestoreSourceBehaviour;
REGISTERED AS    {SWMF.softwareManagement  attribute(7)  futureAutoRestoreSource(13)};
futureAutoRestoreSourceBehaviour BEHAVIOUR
        DEFINED AS
```

"This attribute specifies the source of the information to be restored to the information represented by the managed object instance. The source is either a local managed object or a remote system. The criteria of triggering an automatic restore of information is system specific.";

--A.3.14 identityOfCreator

```
identityOfCreator          ATTRIBUTE
        WITH ATTRIBUTE SYNTAX SWMF.Identity;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR identityOfCreatorBehaviour;
REGISTERED AS    {SWMF.softwareManagement  attribute(7)  identityOfCreator(14)};
identityOfCreatorBehaviour BEHAVIOUR
        DEFINED AS
```

"This attribute identifies the entity that creates the managed object.";

--A.3.15 identityOfLastModifier

```
identityOfLastModifier        ATTRIBUTE
        WITH ATTRIBUTE SYNTAX SWMF.Identity;
        MATCHES FOR     EQUALITY;
        BEHAVIOUR identityOfLastModifierBehaviour;
```

REGISTERED AS {SWMF.softwareManagement attribute(7) identityOfLastModifier(15)};

identityOfLastModifierBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the last, or most recent, modifier of the information represented by the softwareUnit object instance.";

--A.3.16 lastBackupDestination

lastBackupDestination ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.LastBackupDestination;

MATCHES FOR EQUALITY;

BEHAVIOUR lastBackupDestinationBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) lastBackupDestination(16)};

lastBackupDestinationBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the destination, if exists, to which the information represented by the managed object is backed up.";

--A.3.17 lastBackupTime

lastBackupTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.Date;

MATCHES FOR EQUALITY;

BEHAVIOUR lastBackupTimeBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) lastBackupTime(17)};

lastBackupTimeBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the time of the last backup on the information represented by the managed object instance. Valid values for this attribute are ASN.1 GeneralizedTime or NULL (if no backup has been performed on the information).";

--A.3.18 lastRestoreSource

lastRestoreSource ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.LastRestoreSource;

MATCHES FOR EQUALITY;

BEHAVIOUR lastRestoreSourceBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) lastRestoreSource(18)};

lastRestoreSourceBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the source, if exists, from which the information represented by the managed object is restored.";

--A.3.19 lastRestoreTime

lastRestoreTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.Date;

MATCHES FOR EQUALITY;

BEHAVIOUR lastRestoreTimeBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) lastRestoreTime(19)};

lastRestoreTimeBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies the time of the last restore on the information represented by the managed object instance. Valid values for this attribute are ASN.1 GeneralizedTime or NULL (if no restore has been performed on the information).";

--A.3.20 noteField

noteField ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.NoteField;

MATCHES FOR EQUALITY;

BEHAVIOUR noteFieldBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) noteField(20)};

noteFieldBehaviour BEHAVIOUR

DEFINED AS

"This attribute contains any information or comments associated with the managed object, including any specific installation instructions, startup parameters and values, information necessary to activate features of the managed object, etc.";

--A.3.21 softwareDistributorId

softwareDistributorId ATTRIBUTE

WITH ATTRIBUTE SYNTAX SWMF.SimpleNameType;

MATCHES FOR EQUALITY;

BEHAVIOUR softwareDistributorIdBehaviour;

REGISTERED AS {SWMF.softwareManagement attribute(7) softwareDistributorId(21)};

softwareDistributorIdBehaviour BEHAVIOUR

DEFINED AS

"This attribute identifies an instance of the software distributor managed object class.";

--A.4 Name Bindings

--A.4.1 softwareDistributor-subsystem

softwareDistributor-subsystem NAME BINDING

SUBORDINATE OBJECT CLASS softwareDistributor AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS
"Rec. X.723 | ISO/IEC 10165-5":subsystem AND SUBCLASSES;

WITH ATTRIBUTE softwareDistributorId;

CREATE WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE DELETES-CONTAINED-OBJECTS;

REGISTERED AS {SWMF.softwareManagement nameBinding(6) softwareDistributor-subsystem(1)};

--A.4.2 softwareDistributor-system

softwareDistributor-system NAME BINDING

SUBORDINATE OBJECT CLASS softwareDistributor AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS
"Rec. X.721 | ISO/IEC 10165-2:1992":system AND SUBCLASSES;

WITH ATTRIBUTE softwareDistributorId;

CREATE WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE DELETES-CONTAINED-OBJECTS;

REGISTERED AS {SWMF.softwareManagement nameBinding(6) softwareDistributor-system(2)};

--A.4.3 softwareUnit-subsystem

softwareUnit-subsystem NAME BINDING

SUBORDINATE OBJECT CLASS softwareUnit AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS
"Rec. X.723 | ISO/IEC 10165-5":subsystem AND SUBCLASSES;

WITH ATTRIBUTE "Rec. M.3100:1995":softwareId;

CREATE WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE DELETES-CONTAINED-OBJECTS;

REGISTERED AS {SWMF.softwareManagement nameBinding(6) softwareUnit-subsystem(3)};

--A.4.4 softwareUnit-system

softwareUnit-system NAME BINDING

SUBORDINATE OBJECT CLASS softwareUnit AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS
"Rec. X.721 | ISO/IEC 10165-2:1992":system AND SUBCLASSES;

WITH ATTRIBUTE "Rec. M.3100:1995":softwareId;

CREATE WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE DELETES-CONTAINED-OBJECTS;

REGISTERED AS {SWMF.softwareManagement nameBinding(6) softwareUnit-system(4)};

--A.5 Actions

--A.5.1 deliver

deliver ACTION

BEHAVIOUR deliverBehaviour;

MODE CONFIRMED;

PARAMETERS softwareProcessingFailureParameter;

WITH INFORMATION SYNTAX SWMF.DeliverInfo;

REGISTERED AS {SWMF.softwareManagement action(9) deliver(1)};

deliverBehaviour BEHAVIOUR

DEFINED AS

"The deliver service is used by a managing system to request distribution of a software or a set of software. A number of software units may be delivered together to a managed system. If the Deliver Info does not indicate a target destination, the system uses local means to determine the target destination. This service used the CMIS M-ACTION service and procedures defined in ISO/IEC 9595. The Action Type parameter shall indicate deliver.";

--A.5.2 executeProgram

executeProgram ACTION

BEHAVIOUR executeProgramBehaviour;

MODE CONFIRMED;

PARAMETERS softwareProcessingFailureParameter;

WITH INFORMATION SYNTAX SWMF.ExecuteProgramInfo;

WITH REPLY SYNTAX SWMF.ExecuteProgramReply;

REGISTERED AS {SWMF.softwareManagement action(9) executeProgram(2)};

executeProgramBehaviour BEHAVIOUR

DEFINED AS

"The executeProgram service is used by a managing system to request an execution of the program represented by the executableSoftware object instance. A success request will be confirmed with information including the process Id, process owner, and the starting time of the execution.When the executableSoftware is in the busy usageState, any executeProgram request will be denied, until the usageState changes to active or idle. This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC IS 9595.The Action Type parameter shall indicate executeProgram.";

--A.5.3 install

install ACTION

BEHAVIOUR installBehaviour;

MODE CONFIRMED;

PARAMETERS
softwareProcessingFailureParameter;

WITH INFORMATION SYNTAX
SWMF.InstallInfo;

REGISTERED AS {SWMF.softwareManagement action(9) install(3)};

installBehaviour BEHAVIOUR

DEFINED AS

"The install service is used by a managing system to instruct a managed system to install a softwareUnit object instance. If applicable, the install service will effect the automatic update the value of the appliedPatches attribute.This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC 9595.The Action Type parameter shall indicate install.";

--A.5.4 revert

revert ACTION

BEHAVIOUR revertBehaviour;

MODE CONFIRMED;

PARAMETERS
softwareProcessingFailureParameter;

WITH INFORMATION SYNTAX SWMF.RevertArgument;

WITH REPLY SYNTAX SWMF.RevertReply;

REGISTERED AS {SWMF.softwareManagement action(9) revert(4)};

revertBehaviour BEHAVIOUR

DEFINED AS

"The revert service is used by a managing system (e.g. OS) to instruct a managed system to revert an applied patch or set of patches of the software represented by the software unit managed object or the executable software managed object.The revert service will automatically return the value of the appliedPatches attribute of the executableSoftware object instance to which the service is directed.

This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC 9595. The Action Type parameter shall indicate revert patches. The Action Information parameter shall identify the patch or patches to be reverted of the executableSoftware object instance. Each patch identifier is a choice of a system specific identifier (Printable String) or a software unit object instance (Object Instance).";

--A.5.5 terminateValidation

terminateValidation ACTION

BEHAVIOUR terminateValidationBehaviour;

MODE CONFIRMED;

PARAMETERS
softwareProcessingFailureParameter;

WITH INFORMATION SYNTAX SWMF.TerminateValidationArgument;

WITH REPLY SYNTAX SWMF.TerminateValidationReply;

REGISTERED AS {SWMF.softwareManagement action(9) terminateValidation(5)};

terminateValidationBehaviour BEHAVIOUR

DEFINED AS

"The terminateValidation service is used by a managing system to terminate a currently active (running) validation process on one or more objects. Two termination modes are provided for terminating a validation, namely, cancel-mode and truncated-mode. For the truncate-mode, the validation will be terminated and the result of the partially completed validation will be returned in the Action Reply parameter of the terminateValidation M-ACTION. For the cancel-mode, the validation will be terminated and the result of the partially completed validation will be discarded.This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC 9595.The Action Type parameter shall indicate terminateValidation.The Action Information parameter shall indicate the mode of the termination, i.e. cancel-mode or truncate-mode. The Action Reply parameter shall indicate the result of the termination, i.e. terminated (for cancel-mode), terminated with partial result (for truncate-mode), or no active validation to be terminated.";

--A.5.6 validate

validate ACTION

BEHAVIOUR validateBehaviour;

MODE CONFIRMED;

PARAMETERS
softwareProcessingFailureParameter;

WITH INFORMATION SYNTAX SWMF.ValidateInfo;

WITH REPLY SYNTAX SWMF.ValidateReply;

REGISTERED AS {SWMF.softwareManagement action(9) validate(6)};

validateBehaviour BEHAVIOUR

DEFINED AS

"The validate service is used by a managing system to request performing an validation on the information represented by the softwareUnit object instance. This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC 9595. The Action Type parameter shall indicate validate. If the validation process is terminated (by using the terminateValidation M-ACTION) before the normal completion of the validation, then the value 'terminated' (with ASN.1 type NULL) shall be returned in the Action Reply of the validate M-ACTION.";

--A.6 Notifications

--A.6.1 autoBackupReport

autoBackupReport NOTIFICATION

BEHAVIOUR autoBackupReportBehaviour;

WITH INFORMATION SYNTAX SWMF.AutoBackupReportInfo;

REGISTERED AS {SWMF.softwareManagement notification(10) autoBackupReport(1)};

autoBackupReportBehaviour BEHAVIOUR

DEFINED AS

"The autoBackupReport notification is emitted to report an automatic backup of the information represented by this object. The automatic backup criteria and backup destination are specified in the futureAutoBackupTriggerThreshold and futureAutoBackupDestination attributes of the object respectively. The backup destination may be local (i.e. backup to another object of the same class within the local managed system), the managing system, or off-line to a remote system by using a particular file transfer protocol (e.g. FTAM). For local and off-line backup, the result of the backup, i.e. success or failure, will be reported in this notification. For backup to the managing system, the backup information will be in-line included in the notification as a component of the Event Information parameter.";

--A.6.2 autoRestoreReport

autoRestoreReport NOTIFICATION

BEHAVIOUR autoRestoreReportBehaviour;

WITH INFORMATION SYNTAX SWMF.AutoRestoreReportInfo;

REGISTERED AS {SWMF.softwareManagement notification(10) autoRestoreReport(2)};

autoRestoreReportBehaviour BEHAVIOUR

DEFINED AS

"The autoRestoreReport notification is emitted from the managed object when an automatic restore is occurred on the information represented by this object instance. The criteria that triggers the automatic restore is system specific. The source of the restored information and the result of the restore operation (i.e. either success or failure) shall be reported in the notification.";

--A.6.3 deliverResultNotification

deliverResultNotification NOTIFICATION

BEHAVIOUR deliverResultNotificationBehaviour;

WITH INFORMATION SYNTAX SWMF.DeliverResultInfo;

REGISTERED AS {SWMF.softwareManagement notification(10) deliverResultNotification(3)};

deliverResultNotificationBehaviour BEHAVIOUR

DEFINED AS

"This notification type is used to report deliver results and is emitted from the managed object when the delivery is completed. The deliver result parameter may take one of the following values: pass, fail or unknown. This service uses the CMIS M-EVENT-REPORT service and procedures defined in ISO/IEC 9595. The Notification Type parameter shall indicate deliverResult.";

--A.7 Parameters

--A.7.1 alarmEffectOnServiceParameter

alarmEffectOnServiceParameter PARAMETER

CONTEXT EVENT-INFO;

WITH SYNTAX

SWMF.AlarmEffectOnServiceParameter;

BEHAVIOUR alarmEffectOnServiceParameterBehaviour;

REGISTERED AS {SWMF.softwareManagement parameter(5) alarmEffectOnServiceParameter(1)};

alarmEffectOnServiceParameterBehaviour BEHAVIOUR

DEFINED AS

"The alarmEffectOnServiceParameter is a parameter to be included in the ManagementExtension parameter of the AdditionalInformation parameter of the AlarmInfo parameter in an Alarm Reporting service. This parameter indicates whether the service is affected by the alarm. The ManagementExtension is of the form (see Rec. X.721 | ISO/IEC 10165-2):

ManagementExension ::= SEQUENCE {

identifier OBJECT IDENTIFIER,

significance [1] BOOLEAN DEFAULT FALSE,

information [2] ANY DEFINED BY identifier}

The OBJECT IDENTIFIER carried in identifier shall be the value under which this parameter definition is registered. The type carried in information shall be the type identified by the WITH SYNTAX construct of this parameter definition.";

--A.7.2 softwareProcessingFailureParameter

softwareProcessingFailureParameter PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX SWMF.SoftwareProcessingFailure;

BEHAVIOUR softwareProcessingFailureParameterBehaviour;

REGISTERED AS {SWMF.softwareManagement parameter(5) softwareProcessingFailureParameter(2)};

softwareProcessingFailureParameterBehaviour BEHAVIOUR

DEFINED AS

"The softwareProcessingFailureParameter defines the data syntax to be return in a CMIP ProcessingFailure error reply to a M-ACTION if a request for an operation on software is denied due to errors other than those already defined in CMIP-1. The attributes in the returned data syntax include the state attributes of the emitting object. Any other applicable attributes may be included but these are a local matter.";

--A.8 Supporting Productions

SWMF {joint-iso-ccitt ms(9) function(2) part18(18) asn1Module(2) 0}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

IMPORTS

AE-title
FROM
ACSE-1 {joint-iso-ccitt association-control(2) abstract-syntax(1) apdu(0) version(1)}

Attribute, AttributeId, ObjectInstance
FROM
CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

Destination, ManagementExtension, SimpleNameType
FROM
Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1};

--object identifier values--

softwareManagement OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part18(18)}

-- supporting productions

AlarmEffectOnServiceParameter ::= BOOLEAN *-- TRUE = affected --*

AppliedPatches ::= SEQUENCE OF Patch

AutoBackupReportInfo ::= SEQUENCE {
backupResult [0] BackupResult,
additionalInfo [1] SET OF ManagementExtension OPTIONAL }

AutoRestoreReportInfo ::= SEQUENCE {
source AutoRestoreSource,
success BOOLEAN, *-- TRUE for success*
additionalInfo SET OF ManagementExtension OPTIONAL }

AutoRestoreSource ::= CHOICE { localObject ObjectInstance,
remoteSystem GraphicString *-- off-line from remote system --*}

```
BackupResult ::= CHOICE {

        inLine [0]   CHOICE {
            success     BIT STRING,
            fail-pduSizeLimitation      [3]     NULL,
            fail-securityLicensing      [4]     NULL,
            fail-unknown                [5]     NULL}

        local [1]       SEQUENCE {
            destination ObjectInstance, -- in the managed system --
            success BOOLEAN -- TRUE for success --
                },

        offLine [2] SEQUENCE {
            destination GraphicString, --the remote system
            result      CHOICE {
                success                 [6]     NULL,
                fail-securityLicensing  [7]     NULL,
                fail-unknown            [8]     NULL}
                }}

Boolean ::= BOOLEAN

CheckSum ::= BIT STRING

Date ::= CHOICE {
        time GeneralizedTime ,
        noSuchInformation    NULL}

DeliverId ::= CHOICE {
                globalValue     OBJECT IDENTIFIER,
                localValue      INTEGER}

DeliverInfo ::= SEQUENCE {
        deliverId           [0]  DeliverId OPTIONAL,
        -- each delivery may have an id to correlate action and event --
        targetSoftware   SET OF DistributedSoftware,
        targetSystem        [1]  Destination OPTIONAL,
        transferInfo        [2]  TransferInfo OPTIONAL,
        additionalInfo      [3]  SET OF ManagementExtension OPTIONAL }

DeliverResult ::= INTEGER {
        pass (0),
        communicationError (1),
        equipmentError (2),
        qosError (3),
        accessDenied (4),
        notFound (5),
        insufficientSpace (6),
        alreadyDelivered (7),
        inProgress (8),
        unknown (9) }

DeliverResultInfo ::= SEQUENCE {
        deliverId           [0]  DeliverId OPTIONAL,
        deliverResult       [1]  DeliverResult,
        additionalInfo SET OF ManagementExtension OPTIONAL}

DistributedSoftware ::= CHOICE {
        distibutedSoftwareId            GraphicString,
        distributedSoftwarePointer      ObjectInstance }

ExecuteProgramInfo ::= SET OF ManagementExtension

ExecuteProgramReply ::= SEQUENCE {
        processId               INTEGER,
        processOwner    Identity,
        startTime       GeneralizedTime,
        additionalInfo          SET OF ManagementExtension OPTIONAL }

FileLocation ::= SET OF GraphicString -- Empty set means file is not installed --
```

```
FileType ::= INTEGER{
        unstructuredText (0), -- FTAM-1
        unstructuredBinary (1), -- FTAM-3
        blockSpecial (2)}

GlobalTime ::= GeneralizedTime

Identity ::= GraphicString

InformationSize ::=     CHOICE {
        numberOfBits      [0]  INTEGER,
        numberOfBytes     [1]  INTEGER}

Integer ::= INTEGER

InstallInfo ::= SEQUENCE {
        targetSoftware     SET OF DistributedSoftware,
        installInfo        SET OF ManagementExtension }

LastBackupDestination ::= CHOICE {
        notBackedUp NULL,
        localObject ObjectInstance,
        managingSystem AE-title,
        remoteSystem GraphicString}

LastRestoreSource ::= CHOICE {
        notRestored NULL,
        localObject ObjectInstance,
        managingSystem AE-title,
        remoteSystem GraphicString}

NoteField ::= GraphicString

Null ::= NULL

Patch ::= CHOICE {
        patchId    GraphicString, -- system specific identifier --
        patchPointer    ObjectInstance } -- of Software Unit object class --

RevertArgument ::= SEQUENCE {
        revertInfo        [0]  RevertInfo,
        additionalInfo    [1]  SET OF ManagementExtension OPTIONAL }

RevertInfo ::= SEQUENCE OF CHOICE {
        patchId GraphicString, -- system specific identifier --
        patchPointer ObjectInstance } -- Executable Software object class --

RevertReply ::= SEQUENCE {
        revertedPatches    [0]  AppliedPatches,
        additionalInfo     [1]  SET OF ManagementExtension OPTIONAL }

SoftwareProcessingFailure ::= SET OF Attribute        -- state attributes --

TerminateValidationArgument ::= SEQUENCE {
        terminateValidationInfo      [0]  TerminateValidationInfo,
        additionalInfo               [1]  SET OF ManagementExtension OPTIONAL }

TerminateValidationInfo ::= ENUMERATED {
        cancel (0), -- discard the result of the partial audit --
        truncate (1) } -- report the result of the partially completed audit --

TerminateValidationReply ::=  CHOICE {
        noOutStandingValidation     [0]  NULL,
        validationCancelled         [1]  NULL,
        resultOfPartialValidation   [2]  ValidateReply}

TransferInfo ::= SEQUENCE {
        transferProtocol       TransferProtocol,
        protocolSpecificInfo   ANY DEFINED BY transferProtocol OPTIONAL}

TransferProtocol ::= OBJECT IDENTIFIER
```

```
ValidateInfo ::= CHOICE {
        instanceDefaultValidationType   [0] NULL, -- local matter --
        registeredValidationType        [1] OBJECT IDENTIFIER,
        systemSpecificValidationType    [2] SET OF ManagementExtension }

ValidateReply ::= CHOICE {
        validationTerminated        [0] NULL,
        passValidation              [1] NULL,
        passValidationWithResult    [2] SET OF ManagementExtension,
        failValidation              [3] NULL,
        failValidationWithResult    [4] SET OF ManagementExtension }

END     -- end of SWMF supporting productions --
```

--A.9 Backup and Restore Actions--

-- The backup and restore actions are generic actions --
-- and intended for reuse outside the scope of this --
-- Recommendation | International Standard as appropriate --
-- for objects in addition to software objects. --

--A.9.1 backup

```
backup  ACTION
        BEHAVIOUR backupBehaviour;
        MODE CONFIRMED;
        WITH INFORMATION SYNTAX BackupRestoreASN1Module.BackupArgument;
        WITH REPLY SYNTAX BackupRestoreASN1Module.BackupReply;

REGISTERED AS   {SWMF.softwareManagement action(9) backup(7)};

backupBehaviour BEHAVIOUR

        DEFINED AS
```

"The backup service is used by a managing system to request performing a backup on the information represented by the target object instance. This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC 9595.

The Action Type parameter shall indicate backup.

The Action Information parameter shall indicate the destination to which the information will be backed up. Possible destinations are:

- A local managed object of the same class as the one this action is applied to. In this case, the backup operation will be performed internally in the managed system.
- The managing system from which this action is sent. In this case, a copy of the backup information will be sent in-line in the Action Reply.
- A remote system. In this case, the backup information will be transferred off-line to the remote system by using a locally chosen file transfer protocol.

The Action Reply parameter shall indicate the result of the backup. For local or off-line backup, a NULL value indicates the backup is success. For in-line backup, the backup information will be included in the Action Reply parameter.";

--A.9.2 restore--

```
restore ACTION
        BEHAVIOUR restoreBehaviour;
        MODE CONFIRMED;
        WITH INFORMATION SYNTAX BackupRestoreASN1Module.RestoreArgument;
REGISTERED AS   {SWMF.softwareManagement action(9) restore(8)};

restoreBehaviour BEHAVIOUR

        DEFINED AS
```

"The restore service is used by a managing system to request performing a restore on the information represented by the target object instance. This service uses the CMIS M-ACTION service and procedures defined in ISO/IEC 9595.

The Action Type parameter shall indicate restore.

The Action Information parameter shall indicate the source from which the information will be restored. Possible sources are:

- A local managed object of the same class as the one this action is applied to. In this case, the restore operation will be performed internally in the managed system.
- The managing system from which this action is sent. In this case, a copy of the restore information will be sent in-line in the Action Information parameter.
- A remote system. In this case, the restore information will be transferred off-line from the remote system by using a locally chosen file transfer protocol.

The Action Reply parameter shall indicate the result of the restore. A NULL value indicates the restore is success.";

--A.9.3 Backup Restore Supporting Productions--

```
BackupRestoreASN1Module {joint-iso-ccitt ms(9) function(2) part18(18) asn1Module(2) 1}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

IMPORTS

ObjectInstance
FROM
CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

ManagementExtension
FROM
Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1};

--supporting productions--

BackupArgument ::= SEQUENCE {
        backupDestination       [0]  BackupDestination,
        additionalInfo          [1]  SET OF ManagementExtension OPTIONAL }

BackupDestination ::= CHOICE {
        localObject ObjectInstance,
        inLine NULL, -- in-line in the notification in additionalInfo --
        offLine GraphicString -- remote system by, e.g. FTAM --}

BackupReply ::= SEQUENCE {
        reply       [0]  CHOICE {
            success         NULL, -- for local or off-line backup
            inLine          BIT STRING },
        additionalInfo   [1]  SET OF ManagementExtension OPTIONAL }

RestoreArgument ::= SEQUENCE {
        restoreSource       [0]  RestoreSource,
        additionalInfo      [1]  SET OF ManagementExtension OPTIONAL }

RestoreSource ::= CHOICE {
        localObject ObjectInstance,
        inLine BIT STRING       ,
        offLine GraphicString -- remote system via some other transfer protocol, e.g. FTAM --
        }

END     -- of BackupRestoreASN1Module supporting productions –
```

Annex B[3)]

MCS proforma

(This annex forms an integral part of this Recommendation | International Standard)

B.1 Introduction

B.1.1 Purpose and structure

The Management Conformance Summary (MCS) is a statement by a supplier that identifies an implementation and provides information on whether the implementation claims conformance to any of the listed set of documents that specify conformance requirements to OSI management.

The MCS proforma is a document in the form of a questionnaire that when completed by the supplier of an implementation becomes the MCS.

B.1.2 Instructions for completing the MCS proforma to produce an MCS

The supplier of the implementation shall enter an explicit statement in each of the boxes provided. Specific instruction is provided in the text which precedes each table.

B.1.3 Symbols, abbreviations and terms

For all annexes of this Recommendation | International Standard, the following common notations, defined in ITU-T Rec. X.291 and ISO/IEC 9646-2 and ITU-T Rec. X.296 and ISO/IEC 9646-7, are used for the Status column:

m	Mandatory
o	Optional
c	Conditional
x	Prohibited
–	Not applicable or out of scope

NOTES

1 'c', 'm', and 'o' are prefixed by "c:" when nested under a conditional or optional item of the same table;

2 'o' may be suffixed by ".N" (where N is a unique number) for selectable options among a set of status values. Support of at least one of the choices (from the items with the same value of N) is required.

For all annexes of this Recommendation | International Standard, the following common notations, defined in ITU-T Rec. X.291 and ISO/IEC 9646-2 and ITU-T Rec. X.296 and ISO/IEC 9646-7, are used for the Support column:

Y	Implemented
N	Not implemented
–	No answer required
Ig	The item is ignored (i.e. processed syntactically but not semantically)

B.1.4 Table format

Some of the tables in this Recommendation | International Standard have been split because the information is too wide to fit on the page. Where this occurs, the index number of the first block of columns are the index numbers of the corresponding rows of the remaining blocks of columns. A complete table reconstructed from the constituent parts should have the following layout:

Index	First block of columns	Second block of columns	Etc.

3) **Copyright release for MCS proforma**

Users of this Recommendation | International Standard may freely reproduce the MCS proforma in this annex so that it can be used for its intended purpose, and may further publish the completed MCS. Instructions for the MCS proforma are specified in ITU-T Rec. X.724 | ISO/IEC 10165-6.

In this Recommendation | International Standard the constituent parts of the table appear consecutively, starting with the first block of columns.

When a table with subrows is too wide to fit on a page, the continuation table(s) have been constructed with index numbers identical to the index numbers in the corresponding rows of the first table, and with subindex numbers corresponding to the subrows within each indexed row. For example, if Table X.1 has 2 rows and the continuation of Table X.1 has 2 subrows for each row, the tables are presented as follows:

Table X.1 – Title

					Support		
Index	A	B	C	D	E	F	G
1	a	b	–				
2	a	b	–				

Table X.1 *(continued)* – Title

Index	Subindex	H	I	J	K	L
1	1.1	h	i	j		
	1.2	h	i	j		
2	2.1	h	i	j		
	2.1	h	i	j		

A complete table reconstructed from the constituent parts should have the following layout:

					Support							
Index	A	B	C	D	E	F	G	Subindex	H	I	J	K	L
1	a	b	–					1.1	h	i	j		
								1.2	h	i	j		
2	a	b	–					2.1	h	i	j		
								2.2	h	i	j		

References made to cells within tables shall be interpretted as references within reconstructed tables. In the example, above, the reference X.1/1d corresponds with the blank cell in column G for row with Index 1, and X.1/1.2b corresponds with the blank cell in column L for row with Subindex 1.2.

B.2 Identification of the implementation

B.2.1 Date of statement

The supplier of the implementation shall enter the date of this statement in the box below. Use the format DD-MM-YYYY.

Date of statement

B.2.2 Identification of the implementation

The supplier of the implementation shall enter information necessary to uniquely identify the implementation and the system(s) in which it may reside, in the box below.

B.2.3 Contact

The supplier of the implementation shall provide information on whom to contact if there are any queries concerning the contents of the MCS or any referenced implementation conformance statement, in the box below.

B.3 Identification of the Recommendations | International Standards in which the management information is defined

The supplier of the implementation shall enter the title, reference number and date of the publication of the Recommendations | International Standards which specify the management information to which conformance is claimed, in the box below.

Recommendations | International Standards to which conformance is claimed

B.3.1 Technical corrigenda implemented

The supplier of the implementation shall enter the reference numbers of implemented technical corrigenda which modify the identified Recommendations | International Standards, in the box below.

B.3.2 Amendments implemented

The supplier of the implementation shall state the titles and reference numbers of implemented amendments to the identified Recommendation | International Standard, in the box below.

B.4 Management conformance summary

The supplier of the implementation shall state the capabilities and features supported and provide a summary of conformance claims to Recommendations | International Standards using the tables in this annex.

The supplier of the implementation shall specify the roles that are supported, in Table B.1.

Table B.1 – Roles

Index	Roles supported	Status	Support	Additional information
1	Manager role support	o.1		
2	Agent role support	o.1		

The supplier of the implementation shall specify support for the systems management functional units, in Table B.2.

Table B.2 – Systems management functional units

		Manager		Agent		
Index	Systems management functional unit name	Status	Support	Status	Support	Additional information
1	Software control functional unit	c1		c2		
2	Software deliver functional unit	c1		c2		
3	Backup restore functional unit	c1		c2		
c1: if B.1/1a then o else –. c1: if B.1/1a then o else –.						

The supplier of the implementation shall specify support for management information in the manager role, in Table B.3.

Table B.3 – Manager role minimum conformance requirement

Index	Item	Status	Support	Additional information
1	Operations on managed objects	o.2		
2	Backup action for software unit managed object	o.2		
3	Backup action for executable software managed object	o.2		
4	Deliver action for software distributor managed object	o.2		
5	Execute action for executable software managed object	o.2		
6	Install action for software unit managed object	o.2		
7	Install action for executable software managed object	o.2		
8	Restore action for software unit managed object	o.2		
9	Restore action for executable software managed object	o.2		
10	Revert action for software unit managed object	o.2		
11	Revert action for executable software managed object	o.2		
12	Terminate validation for software unit managed object	o.2		
13	Terminate validation for executable software managed object	o.2		
14	Validate action for software unit managed object	o.2		
15	Validate action for executable software managed object	o.2		
16	Auto backup report notification for software objects	o.2		
17	Auto restore report notification for software objects	o.2		
18	Deliver result notification for software distributor	o.2		
19	State change notification for software objects	o.2		

Table B.3 *(concluded)* – Manager role minimum conformance requirement

Index	Item	Status	Support	Additional information
20	Object creation notification for software objects	o.2		
21	Object deletion notification for software objects	o.2		
22	Attribute value change notification for software objects	o.2		
23	Processing error alarm notification for software objects	o.2		
c3: if B.2/1a or B.2/2a then m else (if B.1/1a then o.2 else –).				

The supplier of the implementation shall specify support for management information in the agent role, in Table B.4.

Table B.4 – Agent role minimum conformance requirement

Index	Item	Status	Support	Additional information
1	Software managed object class or executable Software managed object class	c5		
2	Software distributor managed object class	c6		
c5: if B.1/2a and B.2/1a then m else –. c6: if B.1/2a and B.2/2a then m else –.				

Table B.5 – Logging of event records

Index		Status	Support	Additional information
1	Does the implementation support logging of event records in agent role?	o		

The supplier of the implementation shall provide information on claims of conformance to any of the Recommendations | International Standards summarized in the following tables. For each Recommendation | International Standard that the supplier of the implementation claims conformance to, the corresponding conformance statement(s) shall be completed, or referenced by the MCS. The supplier of the implementation shall complete the Support, Table numbers and Additional information columns.

In Tables B.6, B.7, B.8 and B.9, the Status column is used to indicate whether the supplier of the implementation is required to complete the referenced tables or referenced items. Conformance requirements are as specified in the referenced tables or referenced items and are not changed by the value of the MCS Status column. Similarly, the Support column is used by the supplier of the implementation to indicate completion of the referenced tables or referenced items.

Table B.6 – PICS support summary

Index	Identification of the document that includes the PICS proforma	Table numbers of PICS proforma	Description	Constraints and values	Status	Support	Table numbers of PICS	Additional information
1	CCITT Rec. X.730 | ISO/IEC 10164-1	Tables B.2 to B.8	objectCreation, objectDeletion and attributeValueChange	–	m			
2	CCITT Rec. X.730 | ISO/IEC 10164-1	Annex E all tables	SM application context	OBJECT IDENTIFIER	m			
3	CCITT Rec. X.731 | ISO/IEC 10164-2	Tables B.2 to B.4	stateChange	–	m			
4	CCITT Rec. X.733 | ISO/IEC 10164-4	Tables B.2 to B.12	processingErrorAlarm	–	m			
5	CCITT Rec. X.735 | ISO/IEC 10164-6	Annex B all tables	–	–	m			

Table B.7 – MOCS support summary

Index	Identification of the document that includes the MOCS proforma	Table numbers of MOCS proforma	Description	Constraints and values	Status	Support	Table numbers of MOCS	Additional information
1	ITU-T Rec. X.744 | ISO/IEC 10164-18	D.4	softwareUnit		c8			
2	ITU-T Rec. X.744 | ISO/IEC 10164-18	D.5	executableSoftware		c8			
3	ITU-T Rec. X.744 | ISO/IEC 10164-18	D.6	softwareDistributor		c9			
c8: if B.4/1a then o.3 else –. c9: if B.4/2a then m else –.								

Table B.8 – MRCS support summary

Index	Identification of the document that includes the MRCS proforma	Table numbers of MRCS proforma	Description	Constraints and values	Status	Support	Table numbers of MRCS	Additional information
1	ITU-T Rec. X.744 | ISO/IEC 10164-18	Annex E	softwareDistributor-subsystem	–	o			
2	ITU-T Rec. X.744 | ISO/IEC 10164-18	Annex E	softwareDistributor-system	–	o			
3	ITU-T Rec. X.744 | ISO/IEC 10164-18	Annex E	softwareUnit-subsystem	AND SUBCLASSES	o			
4	ITU-T Rec. X.744 | ISO/IEC 10164-18	Annex E	softwareUnit-system	AND SUBCLASSES	o			

Table B.9 – MICS support summary

Index	Identification of the document that includes the MICS proforma	Table numbers of MICS proforma	Description	Constraints and values	Status	Support	Table numbers of MICS	Additional information
1	ITU-T Rec. X.744 | ISO/IEC 10164-18	Tables C.1 and C.2	management operations		c10			
2	ITU-T Rec. X.744 | ISO/IEC 10164-18	Table C.3	notifications		c11			
c10: if B.3/1a or B.3/2a or B.3/3a or B.3/4a or B.3/5a or B.3/6a or B.3/7a or B.3/8a or B.3/9a or B.3/10a or B.3/11a or B.3/12a or B.3/13a or B.3/14a or B.3/15a the m else –. c11: if B.3/16a or B.3/17a or B.3/18a or B.3/19a or B.3/20a or B.3/21a or B.3/22a or B.3/23a then m else –.								

Annex C[4)]

MICS proforma

(This annex forms an integral part of this Recommendation | International Standard)

C.1 Introduction

The purpose of this MICS proforma is to provide a mechanism for a supplier of an implementation which claims conformance, in the manager role, to management information specified in this Recommendation | International Standard, to provide conformance information in a standard form.

C.2 Instructions for completing the MICS proforma to produce an MICS

The MICS proforma contained in this annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.724 | ISO/IEC 10165-6. In addition to the general guidance given in ITU-T Rec. X.724 | ISO/IEC 10165-6, the Additional information column shall be used to identify the object classes for which the management operations are supported. The supplier of the implementation shall state which items are supported in tables below and if necessary, provide additional information.

C.3 Symbols, abbreviations and terms

The following abbreviation is used throughout the MICS proforma:

dmi-att **joint-iso-ccitt ms(9) smi(3) part2(2) attributc(7)**

The notations used for the Status and Support columns are specified in A.1.3.

C.4 Statement of conformance to the management information

C.4.1 Attributes

The specifier of a manager role implementation that claims to support management operations on the attributes specified in this Recommendation | International Standard shall import a copy of Table C.1 and complete it.

Table C.1 – Attribute support

				Set by create		Get		Replace		Add		Remove		Set to default		
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	administrativeState	{dmi-att 31}	ENUMERATED	o.1		o.1		o.1		-		-		o.1		
2	affectedObjectList	{m3100-att 2}	SET OF CHOICE	o.1		o.1		o.1		o.1		o.1		o.1		
3	alarmStatus	{m3100-att 6}	ENUMERATED	o.1		o.1		o.1		-		-		o.1		
4	allomorphs	{dmi-att 50}	SET OF CHOICE	o.1		o.1		-		-		-		-		
5	appliedPatches	{swmf-att 1}	SEQUENCE OF CHOICE	o.1		o.1		o.1		-		-		o.1		

4) **Copyright release for MCS proforma**
Users of this Recommendation | International Standard may freely reproduce the MICS proforma in this annex so that it can be used for its intended purpose, and may further publish the completed MICS.

Table C.1 *(continued)* **– Attribute support**

				Set by create		Get		Replace		Add		Remove		Set to default		
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Addi-tional infor-mation
6	availabilityStatus	{dmi-att 33}	SET OF INTEGER	o.1		o.1		o.1		o.1		o.1		o.1		
7	checkSum	{swmf-att 2}	BIT STRING	o.1		o.1		o.1		-		-		o.1		
8	currentProblemList	{m3100-att 17}	SET OF SEQUENCE	o.1		o.1		o.1		o.1		o.1		o.1		
9	dateDelivered	{swmf-att 3}	CHOICE	o.1		o.1		o.1		-		-		o.1		
10	dateInstalled	{swmf-att 4}	CHOICE	o.1		o.1		o.1		-		-		o.1		
11	dateOfCreation	{swmf-att 5}	GeneralizedTime	o.1		o.1		o.1		-		-		o.1		
12	dateOfLastModification	{swmf-att 6}	CHOICE	o.1		o.1		o.1		-		-		o.1		
13	fileLocation	{swmf-att 7}	CHOICE	o.1		o.1		o.1		-		-		o.1		
14	fileSize	{swmf-att 8}	CHOICE	o.1		o.1		o.1		-		-		o.1		
15	filetType	{swmf-att 9}		o.1		o.1		-		-		-		-		
16	futureAutoBackupDestination	{swmf-att 10}	BackupDestination	o.1		o.1		o.1		-		-		o.1		
17	futureAutoBackupTriggerThreshold	{swmf-att 11}	INTEGER	o.1		o.1		o.1		-		-		o.1		
18	futureAutoRestoreAllowed	{swmf-att 12}	BOOLEAN	o.1		o.1		o.1		-		-		o.1		
19	futureAutoRestoreSource	{swmf-att 13}	CHOICE	o.1		o.1		o.1		-		-		o.1		
20	identityOfCreator	{swmf-att 14}	PrintableString	o.1		o.1		o.1		-		-		o.1		
21	identityOfLastModifier	{swmf-att 15}	PrintableString	o.1		o.1		o.1		-		-		o.1		
22	lastBackupDestination	{swmf-att 16}	CHOICE	o.1		o.1		o.1		-		-		o.1		
23	lastBackupTime	{swmf-att 17}	CHOICE	o.1		o.1		o.1		-		-		o.1		
24	lastRestoreSource	{swmf-att 18}	CHOICE	o.1		o.1		o.1		-		-		o.1		
25	lastRestoreTime	{swmf-att 19}	CHOICE	o.1		o.1		o.1		-		-		o.1		
26	namcBinding	{dmi-att 63}	OBJECT IDENTIFIER	o.1		o.1		-		-		-		-		
27	noteField	{swmf-att 20}	PrintableString	o.1		o.1		o.1		-		-		o.1		
28	objectClass	{dmi-att 65}	CHOICE	o.1		o.1		-		-		-		-		
29	operationalState	{dmi-att 35}	ENUMERATED	-		o.1		-		-		-		-		
30	packages	{dmi-att 66}	SET OF OBJECT IDENTIFIER	o.1		o.1		-		-		-		-		
31	proceduralStatus	{dmi-att 36}	SET OF INTEGER	o.1		o.1		-		-		-		-		
32	softwareDistributorId	[swmf-att 21}	SimpleNameType	o.1		o.1		-		-		-		-		
33	softwareId	{m3100-att 38}	CHOICE	o.1		o.1		-		-		-		-		

Table C.1 *(concluded)* **– Attribute support**

				Set by create		Get		Replace		Add		Remove		Set to default		
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
34	usageState	{dmi-att 39}	ENUMERATED	-		o.1		-		-		-		-		
35	userLabel	{m3100-att 50}	GraphicString	o.1		o.1		o.1		-		-		-		
36	vendorName	{m3100-att 51}	GraphicString	o.1		o.1		o.1		-		-		-		
37	version	{m3100-att 52}	GraphicString	o.1		o.1		o.1		-		-		-		

C.4.2 Actions

The specifier of a manager role implementation that claims to support actions specified in this Recommendation | International Standard shall import a copy of Table C.2 and complete it.

Table C.2 – Action support summary

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	backup	{swmf-act 7}	softwareProcessingFailureParameter	o.1			1.1	BackupArgument	Information Syntax SEQUENCE	c:m		
							1.1.1	restoreSource	CHOICE	c:m		
							1.1.1.1	localObject	Object Instance	c:o.1		
							1.1.1.2	inLine	BIT STRING	c:o.1		
							1.1.1.3	offLine	GraphicString	c:o.1		
							1.1.2	additionalInfo	SET OF ANY	c:o		
							1.2	BackupReply	Reply Syntax SEQUENCE	c:m		
							1.2.1	reply	CHOICE	c:m		
							1.2.1.1	success	NULL	c:o.2		
							1.2.1.2	inLine	BIT STRING	c:o.2		
							1.2.2	additionalInfo	SET OF ANY	c:o		
2	executeProgram	{swmf-act 2}	softwareProcessingFailureParameter	o.1			2.1	executeProgramInfo	Information Syntax SET OF SEQUENCE	c:m		
							2.1.2	significance	BOOLEAN	c:o		
							2.1.3	information	ANY DEFINED BY identifier	c:m		
							2.2	ExecuteProgramReply	Reply Syntax SEQUENCE	c:m		
							2.2.1	processId	INTEGER	c:m		
							2.2.2	processOwner	GraphicString	c:m		

Table C.2 *(continued)* – **Action support summary**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Sta-tus	Sup-port	Addi-tional infor-mation	Subindex	Action field name label	Constraints and values	Status	Sup-port	Addi-tional infor-mation
							2.2.3	startTime	GeneralizedTime	c:m		
							2.2.4	additionalInfo	SET OF SEQUENCE	c:o		
							2.2.4.1	identifier	OBJECT IDENTIFIER	c:m		
							2.2.4.2	significance	BOOLEAN	c:o		
							2.2.4.3	information	ANY DEFINED BY identifier	c:m		
3	install	{swmf-act 3}	softwareProcessingFailureParameter	o.1			3.1	InstallInfo	Information Syntax SEQUENCE	c:m		
							3.1.1	targetSoftware	SET OF CHOICE	c:m		
							3.1.1.1	distributedSoftwareId	GraphicString	c:o.1		
							3.1.1.2	distributedSoftwarePointer	CHOICE	c:o.1		
							3.1.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.2		
							3.1.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							3.1.1.2.1.2	AttributeValue	ANY	c:m		
							3.1.1.2.2	nonSpecificForm	OCTET STRING	c:o.2		
							3.1.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.2		
							3.1.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							3.1.1.2.3.2	AttributeValue	ANY	c:m		
							3.1.2	installInfo	SET OF SEQUENCE	c:m		
							3.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							3.1.2.2	significance	BOOLEAN	c:o		
							3.1.2.3	information	ANY DEFINED BY identifier	c:m		
4	restore	{swmf-act 8}		o.1			4.1	RestoreArgument	Information Syntax SEQUENCE	c:m		
							4.1.1	restoreSource	CHOICE	c:m		
							4.1.1.1	localObject	Object Instance	c:o.1		
							4.1.1.2	inLine	BIT STRING	c:o.1		
							4.1.1.3	offLine	Printable String	c:o.1		

Table C.2 *(continued)* – **Action support summary**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							4.1.2	additionalInfo	SET OF ANY	c:o		
5	revert	{swmf-act 4}	softwareProcessingFailureParameter	o.1			5.1	RevertArgument	Information Syntax SEQUENCE	c:m		
							5.1.1	revertInfo	SEQUENCE OF CHOICE	c:m		
							5.1.1.1	patchId	GraphicString	c:o.3		
							5.1.1.2	patchPointer	CHOICE	c:o.3		
							5.1.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.4		
							5.1.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							5.1.1.2.1.2	AttributeValue	ANY	c:m		
							5.1.1.2.2	nonSpecificForm	OCTET STRING	c:o.4		
							5.1.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.4		
							5.1.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							5.1.1.2.3.2	AttributeValue	ANY	c:m		
							5.1.2	additionalInfo	SET OF SEQUENCE	c:o		
							5.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							5.1.2.2	significance	BOOLEAN	c:o		
							5.1.2.3	information	ANY DEFINED BY identifier	c:m		
							5.2	RevertReply	Reply Syntax SEQUENCE	c:m		
							5.2.1	appliedPatches	SEQUENCE OF CHOICE	c:m		
							5.2.1.1	patchId	GraphicString	c:o.5		
							5.2.1.2	patchPointer	CHOICE	c:o.5		
							5.2.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.6		
							5.2.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							5.2.1.2.1.2	AttributeValue	ANY	c:m		
							5.2.1.2.2	nonSpecificForm	OCTET STRING	c:o.6		
							5.2.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.6		
							5.2.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		

Table C.2 *(continued)* – **Action support summary**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							5.2.1.2.3.2	AttributeValue	ANY	c:m		
							5.2.2	additionalInfo	SET OF SEQUENCE	c:o		
							5.2.2.1	identifier	OBJECT IDENTIFIER	c:m		
							5.2.2.2	significance	BOOLEAN	c:o		
							5.2.2.3	information	ANY DEFINED BY identifier	c:m		
6	terminateValidation	{swmf-act 5}	softwareProcessingFailureParameter	o.1			6.1	TerminateValidationArgument	Information Syntax SEQUENCE	c:m		
							6.1.1	terminateValidationInfo	ENUMERATED	c:m		
							6.1.2	additionalInfo	SET OF SEQUENCE	c:o		
							6.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							6.1.2.2	significance	BOOLEAN	c:o		
							6.1.2.3	information	ANY DEFINED BY identifier	c:m		
							6.2	TerminateValidationReply	Reply Syntax CHOICE	c:m		
							6.2.1	noOutStandingValidation	NULL	c:o.7		
							6.2.2	validationCancelled	NULL	c:o.7		
							6.2.3	resultOfPartialValidation	CHOICE	c:o.7		
							6.2.3.1	validationTerminated	NULL	c:o.8		
							6.2.3.2	passValidation	NULL	c:o.8		
							6.2.3.3	passValidationWithResult	SET OF SEQUENCE	c:o.8		
							6.2.3.3.1	identifier	OBJECT IDENTIFIER	c:m		
							6.2.3.3.2	significance	BOOLEAN	c:o		
							6.2.3.3.3	information	ANY DEFINED BY identifier	c:m		
							6.2.3.4	failValidation	NULL	c:o.8		
							6.2.3.5	failValidationWithResult	SET OF SEQUENCE	c:o.8		
							6.2.3.5.1	identifier	OBJECT IDENTIFIER	c:m		
							6.2.3.5.2	significance	BOOLEAN	c:o		

Table C.2 *(continued)* **– Action support summary**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							6.2.3.5.3	information	ANY DEFINED BY identifier	c:m		
7	validate	{swmf-act 6}	softwareProcessingFailureParameter	o.1			7.1	ValidateInfo	Information Syntax CHOICE	c:m		
							7.1.1	instanceDefaultValidationTypc	NULL	c:o.9		
							7.1.2	registeredValidationType	OBJECT IDENTIFIER	c:o.9		
							7.1.3	systemSpecificValidationType	SET OF SEQUENCE	c:o.9		
							7.1.3.1	identifier	OBJECT IDENTIFIER	c:m		
							7.1.3.2	significance	BOOLEAN	c:o		
							7.1.3.3	information	ANY DEFINED BY identifier	c:m		
							7.2	ValidateReply	Reply Syntax CHOICE	c:m		
							7.2.1	validationTerminated	NULL	c:o.10		
							7.2.2	passValidation	NULL	c:o.10		
							7.2.3	passValidationWithResult	SET OF SEQUENCE	c:o.10		
							7.2.3.1	identifier	OBJECT IDENTIFIER	c:m		
							7.2.3.2	significance	BOOLEAN	c:o		
							7.2.3.3	information	ANY DEFINED BY identifier	c:m		
							7.2.4	failValidation	NULL	c:o.10		
							7.2.5	failValidationWithResult	SET OF SEQUENCE	c:o.10		
							7.2.5.1	identifier	OBJECT IDENTIFIER	c:m		
							7.2.5.2	significance	BOOLEAN	c:o		
							7.2.5.3	information	ANY DEFINED BY identifier	c:m		
8	deliver	{swmf-act 1}	softwareProcessingFailureParameter	o.1			8.1	DeliverInfo	Information Syntax SEQUENCE	c:m		
							8.1.1	deliverId	CHOICE	o		
							8.1.1.1	globalValue	OBJECT IDENTIFIER	c:o.1		
							8.1.1.2	localValue	INTEGER	c:o.1		
							8.1.2	targetSoftware	SET OF CHOICE	m		

Table C.2 *(concluded)* **– Action support summary**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							8.1.2.1	distributedSoftwareId	GraphicString	o.2		
							8.1.2.2	distributedSoftwarePointer	CHOICE	o.2		
							8.1.2.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.3		
							8.1.2.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							8.1.2.2.1.2	AttributeValue	ANY	c:m		
							8.1.2.2.2	nonSpecificForm	OCTET STRING	c:o.3		
							8.1.2.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.3		
							8.1.2.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							8.1.2.2.3.2	AttributeValue	ANY	c:m		
							8.1.3	targetSystem	CHOICE	o		
							8.1.3.1	single	AE-title	c:o.4		
							8.1.3.2	multiple	SET OF AE-title	c:o.4		
							8.1.4	transferInfo	SEQUENCE	o		
							8.1.4.1	transferProtocol	CHOICE	c:m		
							8.1.4.1.1	globalValue	OBJECT IDENTIFIER	c:o.5		
							8.1.4.1.2	localValue	INTEGER	c:o.5		
							8.1.4.2	protocolSpecificInfo	ANY DEFINED BY transferProtocol	c:o		
							8.1.5	additionalInfo	SET OF SEQUENCE	o		
							8.1.5.1	identifier	OBJECT IDENTIFIER	c:m		
							8.1.5.2	significance	BOOLEAN	c:o		
							8.1.5.3	information	ANY DEFINED BY identifier	c:m		

C.4.3 Notifications

The specifier of a manager role implementation that claims to support notifications specified in this Recommendation | International Standard shall import a copy of Table C.3 and complete it.

Table C.3 – Notification support summary

					Support									
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	attributeValueChange	{dmi-not1}		o.1				1.1	AttributeValueChangeInfo		Information Syntax SEQUENCE	c:m		
								1.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								1.1.2	attributeIdentifierList	{dmi-att 8}	SET OF CHOICE	c:o		
								1.1.2.1	globalForm	-	OBJECT IDENTIFIER	c:o.1		
								1.1.2.2	localForm		INTEGER	c:o.1		
								1.1.3	attributeValueChangeDefinition	{dmi-att 10}	SET OF SEQUENCE	c:m		
								1.1.3.1	attributeID	-	CHOICE	c:m		
								1.1.3.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.2		
								1.1.3.1.2	localForm	-	INTEGER	c:o.2		
								1.1.3.2	oldAttributeValue	-	ANY DEFINED BY attributeID	c:o		
								1.1.3.3	newAttributeValue	-	ANY DEFINED BY attributeID	c:m		
								1.1.4	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								1.1.5	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								1.1.5.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								1.1.5.2	sourceObjectInst	-	CHOICE	c:o		
								1.1.5.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.3		
								1.1.5.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								1.1.5.2.1.2	AttributeValue	-	ANY	c:m		
								1.1.5.2.2	nonSpecificForm	-	OCTET STRING	c:o.3		
								1.1.5.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.3		
								1.1.5.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								1.1.5.2.3.2	AttributeValue	-	ANY	c:m		

Table C.3 *(continued)* **– Notification support summary**

					Support										
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								1.1.6	additionalText	{dmi-att 7}	GraphicString	c:o		
								1.1.7	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								1.1.7.1	identifier	-	OBJECT IDENTIFIER	c:m		
								1.1.7.2	significance	-	BOOLEAN	c:o		
								1.1.7.3	information	-	ANY DEFINED BY identifier	c:m		
2	autoBackupReport	{2 9 2 18 10 1}		o.1				2.1	AutoBackupReportInfo		Information Syntax SEQUENCE	c:m		
								2.1.1	backupResult	-	CHOICE	c:m		
								2.1.1.1	inLine	-	CHOICE	c:o.4		
								2.1.1.2	local	-	SEQUENCE	c:o.4		
								2.1.1.2.1	destination	-	CHOICE	c:m		
								2.1.1.2.1.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.5		
								2.1.1.2.1.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								2.1.1.2.1.1.2	AttributeValue	-	ANY	c:m		
								2.1.1.2.1.2	nonSpecificForm	-	OCTET STRING	c:o.5		
								2.1.1.2.1.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.5		
								2.1.1.2.1.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								2.1.1.2.1.3.2	AttributeValue	-	ANY	c:m		
								2.1.1.2.2	success	-	BOOLEAN	c:m		
								2.1.1.3	offLine	-	SEQUENCE	c:o.4		
								2.1.1.3.1	destination	-	GraphicString	c:m		
								2.1.1.3.2	result	-	CHOICE	c:m		
3	autoRestoreReport	{2 9 2 18 10 2}		o.1				3.1	AutoRestoreReportInfo		Information Syntax SEQUENCE	c:m		
								3.1.1	source	-	CHOICE	c:m		
								3.1.1.1	localObject	-	CHOICE	c:o.6		
								3.1.1.1.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.7		

Table C.3 *(continued)* – **Notification support summary**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								3.1.1.1.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								3.1.1.1.1.2	AttributeValue	-	ANY	c:m		
								3.1.1.1.2	nonSpecificForm	-	OCTET STRING	c:o.7		
								3.1.1.1.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.7		
								3.1.1.1.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								3.1.1.1.3.2	AttributeValue	-	ANY	c:m		
								3.1.1.2	remoteSystem	-	GraphicString	c:o.6		
								3.1.2	success	-	BOOLEAN	c:m		
								3.1.3	additionalInfo	-	SET OF SEQUENCE	c:o		
								3.1.3.1	identifier	-	OBJECT IDENTIFIER	c:m		
								3.1.3.2	significance	-	BOOLEAN	c:o		
								3.1.3.3	information	-	ANY DEFINED BY identifier	c:m		
4	objectCreation	{dmi-not6}		o.1				4.1	ObjectInfo		Information Syntax SEQUENCE	c:m		
								4.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								4.1.2	attributeList	{dmi-att 9}	SET OF SEQUENCE	c:o		
								4.1.2.1	attributeId	-	CHOICE	c:m		
								4.1.2.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.8		
								4.1.2.1.2	localForm	-	INTEGER	c:o.8		
								4.1.2.2	attributeValue	-	ANY DEFINED BY attributeId	c:m		
								4.1.3	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								4.1.4	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								4.1.4.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								4.1.4.2	sourceObjectInst	-	CHOICE	c:o		

Table C.3 *(continued)* **– Notification support summary**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								4.1.4.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.9		
								4.1.4.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								4.1.4.2.1.2	AttributeValue	-	ANY	c:m		
								4.1.4.2.2	nonSpecificForm	-	OCTET STRING	c:o.9		
								4.1.4.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.9		
								4.1.4.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								4.1.4.2.3.2	AttributeValue	-	ANY	c:m		
								4.1.5	additionalText	{dmi-att 7}	GraphicString	c:o		
								4.1.6	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								4.1.6.1	identifier	-	OBJECT IDENTIFIER	c:m		
								4.1.6.2	significance	-	BOOLEAN	c:o		
								4.1.6.3	information	-	ANY DEFINED BY identifier	c:m		
5	objectDeletion	{dmi-not7}		o.1				5.1	ObjectInfo		Information Syntax SEQUENCE	c:m		
								5.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								5.1.2	attributeList	{dmi-att 9}	SET OF SEQUENCE	c:o		
								5.1.2.1	attributeId	-	CHOICE	c:m		
								5.1.2.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 10		
								5.1.2.1.2	localForm	-	INTEGER	c:o. 10		
								5.1.2.2	attributeValue	-	ANY DEFINED BY attributeId	c:m		
								5.1.3	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								5.1.4	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								5.1.4.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		

Table C.3 *(continued)* – **Notification support summary**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								5.1.4.2	sourceObjectInst	-	CHOICE	c:o		
								5.1.4.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11		
								5.1.4.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								5.1.4.2.1.2	AttributeValue	-	ANY	c:m		
								5.1.4.2.2	nonSpecificForm	-	OCTET STRING	c:o. 11		
								5.1.4.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11		
								5.1.4.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								5.1.4.2.3.2	AttributeValue	-	ANY	c:m		
								5.1.5	additionalText	{dmi-att 7}	GraphicString	c:o		
								5.1.6	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								5.1.6.1	identifier	-	OBJECT IDENTIFIER	c:m		
								5.1.6.2	significance	-	BOOLEAN	c:o		
								5.1.6.3	information	-	ANY DEFINED BY identifier	c:m		
6	processingErrorAlarm	{dmi-not10}		o.1			alarmEffectOnServiceParameter	6.1	AlarmInfo		Information Syntax SEQUENCE	c:m		
								6.1.1	probableCause	{dmi-att 18}	CHOICE	c:m		
								6.1.1.1	globalValue	-	OBJECT IDENTIFIER	o.12		
								6.1.1.2	localValue	-	INTEGER	o.12		
								6.1.2	specificProblems	{dmi-att 27}	SET OF CHOICE	o		
								6.1.2.1	OBJECT IDENTIFIER	-	OBJECT IDENTIFIER	c:o. 13		
								6.1.2.2	INTEGER	-	INTEGER	c:o. 13		
								6.1.3	perceivedSeverity	{dmi-att 17}	ENUMERATED	c:m		

Table C.3 *(continued)* **– Notification support summary**

					Support									
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								6.1.4	backedUpStatus	{dmi-att 11}	BOOLEAN	o		
								6.1.5	backUpObject	{dmi-att 40}	CHOICE	o		
								6.1.5.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 14		
								6.1.5.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.5.1.2	AttributeValue	-	ANY	c:m		
								6.1.5.2	nonSpecificForm	-	OCTET STRING	c:o. 14		
								6.1.5.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 14		
								6.1.5.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.5.3.2	AttributeValue	-	ANY	c:m		
								6.1.6	trendIndication	{dmi-att 30}	ENUMERATED	o		
								6.1.7	thresholdInfo	{dmi-att 29}	SEQUENCE	o		
								6.1.7.1	triggeredThreshold	-	CHOICE	c:m		
								6.1.7.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 15		
								6.1.7.1.2	localForm	-	INTEGER	c:o. 15		
								6.1.7.2	observedValue	-	CHOICE	c:m		
								6.1.7.2.1	integer	-	INTEGER	c:o. 16		
								6.1.7.2.2	real	-	REAL	c:o. 16		
								6.1.7.3	thresholdLevel	-	CHOICE	c:o		
								6.1.7.3.1	up	-	SEQUENCE	c:o. 17		
								6.1.7.3.1.1	high	-	CHOICE	c:m		
								6.1.7.3.1.1.1	integer	-	INTEGER	c:o. 18		
								6.1.7.3.1.1.2	real	-	REAL	c:o. 18		

Table C.3 *(continued)* – **Notification support summary**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
					Confirmed	Non-confirmed								
								6.1.7.3.1.2	low	-	CHOICE	c:o		
								6.1.7.3.1.2.1	integer	-	INTEGER	c:o. 19		
								6.1.7.3.1.2.2	real	-	REAL	c:o. 19		
								6.1.7.3.2	down	-	SEQUENCE	c:o. 17		
								6.1.7.3.2.1	high	-	CHOICE	c:m		
								6.1.7.3.2.1.1	integer	-	INTEGER	c:o. 20		
								6.1.7.3.2.1.2	real	-	REAL	c:o. 20		
								6.1.7.3.2.2	low	-	CHOICE	c:m		
								6.1.7.3.2.2.1	integer	-	INTEGER	c:o. 21		
								6.1.7.3.2.2.2	real	-	REAL	c:o. 21		
								6.1.7.4	armTime	-	GeneralizedTime	c:o		
								6.1.8	notificationIdentifier	{dmi-att 16}	INTEGER	o		
								6.1.9	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	o		
								6.1.9.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								6.1.9.2	sourceObjectInst	-	CHOICE	c:o		
								6.1.9.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 22		
								6.1.9.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.9.2.1.2	AttributeValue	-	ANY	c:m		
								6.1.9.2.2	nonSpecificForm	-	OCTET STRING	c:o. 22		
								6.1.9.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 22		
								6.1.9.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.9.2.3.2	AttributeValue	-	ANY	c:m		
								6.1.10	stateChangeDefinition	{dmi-att 28}	SET OF SEQUENCE	o		

Table C.3 *(continued)* **– Notification support summary**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								6.1.10.1	attributeID	-	CHOICE	c:m		
								6.1.10.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 23		
								6.1.10.1.2	localForm	-	INTEGER	c:o. 23		
								6.1.10.2	oldAttributeValue	-	ANY DEFINED BY attributeID	c:o		
								6.1.10.3	newAttributeValue	-	ANY DEFINED BY attributeID	c:m		
								6.1.11	monitoredAttributes	{dmi-att 15}	SET OF SEQUENCE	o		
								6.1.11.1	attributeId	-	CHOICE	c:m		
								6.1.11.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 24		
								6.1.11.1.2	localForm	-	INTEGER	c:o. 24		
								6.1.11.2	attributeValue	-	ANY DEFINED BY attributeId	c:m		
								6.1.12	proposedRepairActions	{dmi-att 19}	SET OF CHOICE	o		
								6.1.12.1	OBJECT IDENTIFIER	-	OBJECT IDENTIFIER	c:o. 25		
								6.1.12.2	INTEGER	-	INTEGER	c:o. 25		
								6.1.13	additionalText	{dmi-att 7}	GraphicString	o		
								6.1.14	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	o		
								6.1.14.1	identifier	-	OBJECT IDENTIFIER	c:m		
								6.1.14.2	significance	-	BOOLEAN	c:o		
								6.1.14.3	information	-	ANY DEFINED BY identifier	c:m		
7	stateChange	{dmi-not14}		o.1				7.1	StateChangeInfo		Information Syntax SEQUENCE	c:m		
								7.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								7.1.2	attributeIdentifierList	{dmi-att 8}	SET OF CHOICE	c:o		
								7.1.2.1	globalForm	-	OBJECT IDENTIFIER	c:o. 26		

Table C.3 *(continued)* – **Notification support summary**

					Support									
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								7.1.2.2	localForm	-	INTEGER	c:o. 26		
								7.1.3	stateChangeDefinition	{dmi-att 28}	SET OF SEQUENCE	c:m		
								7.1.3.1	attributeID	-	CHOICE	c:m		
								7.1.3.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 27		
								7.1.3.1.2	localForm	-	INTEGER	c:o. 27		
								7.1.3.2	oldAttributeValue	-	ANY DEFINED BY attributeID	c:o		
								7.1.3.3	newAttributeValue	-	ANY DEFINED BY attributeID	c:m		
								7.1.4	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								7.1.5	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								7.1.5.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								7.1.5.2	sourceObjectInst	-	CHOICE	c:o		
								7.1.5.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 28		
								7.1.5.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								7.1.5.2.1.2	AttributeValue	-	ANY	c:m		
								7.1.5.2.2	nonSpecificForm	-	OCTET STRING	c:o. 28		
								7.1.5.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 28		
								7.1.5.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								7.1.5.2.3.2	AttributeValue	-	ANY	c:m		
								7.1.6	additionalText	{dmi-att 7}	GraphicString	c:o		
								7.1.7	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		

Table C.3 *(concluded)* **– Notification support summary**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								7.1.7.1	identifier	-	OBJECT IDENTIFIER	c:m		
								7.1.7.2	significance	-	BOOLEAN	c:o		
								7.1.7.3	information	-	ANY DEFINED BY identifier	c:m		
8	deliverResultNotification	{swmf-not 3}		o.1				8.1	DeliverResultInfo		Information Syntax SEQUENCE	c:m		
								8.1.1	deliverId	-	CHOICE	o		
								8.1.1.1	globalValue	-	OBJECT IDENTIFIER	c:o.1		
								8.1.1.2	localValue	-	INTEGER	c:o.1		
								8.1.2	deliverResult	-	ENUMERATED	m		
								8.1.3	additionalInfo	-	SET OF SEQUENCE	o		
								8.1.3.1	identifier	-	OBJECT IDENTIFIER	c:m		
								8.1.3.2	significance	-	BOOLEAN	c:o		
								8.1.3.3	information	-	ANY DEFINED BY identifier	c:m		

C.4.4 Create and delete management operations

The specifier of a manager role implementation that claims to support the create or delete management operations on the managed objects specified in this Recommendation | International Standard shall import a copy of this table and complete it.

Table C.4 – Create and delete support

Index	Operation	Constraints and values	Status	Support	Additional information
1	Create support	softwareUnit, executableSoftware or softwareDistributor	o.1		
1.1	Create with reference object	–	c:o		
2	Delete support	softwareUnit, executableSoftware or softwareDistributor	o.1		

Annex D[5)]

MOCS proforma

(This annex forms an integral part of this Recommendation | International Standard)

D.1 Introduction

The purpose of this MOCS proforma is to provide a mechanism for a supplier of an implementation which claims conformance, in the agent role, to a managed object class, to provide conformance information in a standard form.

D.2 Instructions for completing the MOCS proforma to produce a MOCS

The MOCS proforma contained in this annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.724 | ISO/IEC 10165-6. The supplier of the implementation shall state which items are supported in the tables below and if necessary, provide additional information.

D.3 Symbols, abbreviations and terms

The following abbreviations are used throughout the MOCS proforma:

dmi-att **joint-iso-ccitt ms(9) smi(3) part2(2) attribute(7)**
dmi-nb **joint-iso-ccitt ms(9) smi(3) part2(2) nameBinding(6)**
dmi-not **joint-iso-ccitt ms(9) smi(3) part2(2) notification(10)**
dmi-obj **joint-iso-ccitt ms(9) smi(3) part2(2) managedObjectClass(3)**
dmi-pkg **joint-iso-ccitt ms(9) smi(3) part2(2) package(4)**

The notations used for the Status and Support columns are specified in A.1.3.

D.4 Software unit managed object class

D.4.1 Statement of conformance to the managed object class

The supplier of the implementation shall state whether or not all mandatory features of the software unit managed object class are supported, and if the actual class supported is the same as the managed object class to which conformance is claimed.

Table D.1 – Software Unit Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	softwareUnit	{swmf-obj 1}		

If the answer to the actual class question in Table D.1 is "N", the supplier of the implementation shall supply the actual class support details in Table D.3.

Table D.2 – Software Unit Actual class support

Index	Managed object class template for actual class	Value of object identifier for managed object class definition of actual class	Additional information

5) **Copyright release for MOCS proforma**
Users of this Recommendation | International Standard may freely reproduce the MOCS proforma in this annex so that it can be used for its intended purpose, and may further publish the completed MOCS. Instructions for completing the MOCS proforma are specified in ITU-T Rec. X.724 | ISO/IEC 10165-6.

D.4.2 Packages

Table D.3 – Software Unit Package support

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	administrativeOperationalStatesPackage	{m3100-pkg 1}	Mandatory	m		
2	affectedObjectListPackage	{m3100-pkg 2}	"an instance supports it"	o		
3	allomorphicPackage	{dmi-pkg 17}	"if an object supports allomorphism"	c1		
4	appliedPatchPackage	{swmf-pkg 1}	"an instance supports software patching"	o		
5	attributeValueChangeNotificationPackage	{m3100-pkg 4}	"the attributeValueChange notification defined in Recommendation X.721 is supported by an instance of this class."	o		
6	checkSumPackage	{swmf-pkg 2}	"an instance supports it"	o		
7	createDeleteNotificationsPackage	{m3100-pkg 10}	"the objectCreation and objectDeletion notifications defined in Recommendation X.721 are supported by an instance of this class."	o		
8	currentProblemListPackage	{m3100-pkg 13}	"an instance supports it"	o		
9	fileInformationPackage	{swmf-pkg 4}	"an instance supports it"	o		
10	filePackage	{swmf-pkg 5}	"an instance supports it"	o		
11	informationAutoBackupPackage	{swmf-pkg 6}	"an instance supports it"	o		
12	informationAutoRestorePackage	{swmf-pkg 7}	"an instance supports it"	o		
13	informationBackupPackage	{swmf-pkg 8}	"an instance supports it"	o		
14	informationRestorePackage	{swmf-pkg 9}	"an instance supports it"	o		
15	installPackage	{swmf-pkg 10}	"an instance supports it"	o		
16	noteFieldPackage	{swmf-pkg 11}	"an instance supports it"	o		
17	packagesPackage	{dmi-pkg 16}	"any registered package, other than this package has been instantiated."	m		
18	processingErrorAlarmOnServicePackage	{swmf-pkg 12}	Mandatory	m		
19	revertPackage	{swmf-pkg 13}	"an instance supports it"	o		
20	softwarePackage		Mandatory	m		
21	softwareProcessingErrorAlarmPackage	{m3100-pkg 26}	"an instance supports it"	o		
22	softwareUnitPackage		Mandatory	m		
23	stateChangeNotificationPackage	{m3100-pkg 28}	"the stateChange notification defined in Recommendation X.721 is supported by an instance of this class."	o		
24	terminateValidationPackage	{swmf-pkg 14}	"the validationPackage is present and an instance supports it."	c2		
25	topPackage		Mandatory	m		
26	usageStatePackage	{swmf-pkg 15}	"an instance supports it"	o		
27	userLabelPackage	{m3100-pkg 32}	"an instance supports it"	o		
28	validationPackage	{swmf-pkg 16}	"an instance supports it"	o		
29	vendorNamePackage	{m3100-pkg 33}	"an instance supports it"	o		
30	versionPackage	{m3100-pkg 34}	"an instance supports it"	o		

c1: if D.41/1b then – else m.
c2: if D.4.3/28a then o else –.

D.4.3 Attributes

Table D.4 – Software Unit Attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get		Replace		Add		Remove		Set to default		Addi-tional infor-mation
				Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	
1	administrativeState	{dmi-att 31}	ENUMERA-TED	m		m		m		-		-		c3		
2	affectedObjectList	{m3100-att 2}	SET OF CHOICE	c4		c4a		c4		c4		c4		c4		
3	alarmStatus	{m3100-att 6}	ENUMERA-TED	c5		c5a		c5		-		-		c5		
4	allomorphs	{dmi-att 50}	SET OF CHOICE	c6		c7		-		-		-		-		
5	appliedPatches	{swmf-att 1}	SEQUENCE OF CHOICE	c8		c9		c8		-		-		c8		
6	availabilityStatus	{dmi-att 33}	SET OF INTEGER	c3		m		c3		c3		c3		c3		
7	checkSum	{swmf-att 2}	BIT STRING	c10		c11		c10		-		-		c10		
8	currentProblemList	{m3100-att 17}	SET OF SEQUENCE	c12		c13		c12		c12		c12		c12		
9	dateDelivered	{swmf-att 3}	CHOICE	c14		c15		c14		-		-		c14		
10	dateInstalled	{swmf-att 4}	CHOICE	c16		c17		c16		-		-		c16		
11	dateOfCreation	{swmf-att 5}	GeneralizedTime	c18		c19		c18		-		-		c18		
12	dateOfLastModification	{swmf-att 6}	CHOICE	c20		c21		c20		-		-		c20		
13	fileLocation	{swmf-att 7}	CHOICE	c22		c23		c22		-		-		c22		
14	fileSize	{swmf-att 8}	CHOICE	c24		c25		c24		-		-		c24		
15	filetType	{swmf-att 9}		o		o		-		-		-		-		
16	futureAutoBackupDestination	{swmf-att 10}	BackupDestination	c26		c26		c26		-		-		c27		
17	futureAutoBackupTriggerThreshold	{swmf-att 11}	INTEGER	c26		c26		c26		-		-		c27		
18	futureAutoRestoreAllowed	{swmf-att 12}	BOOLEAN	c28		c28		c28		-		-		c29		
19	futureAutoRestoreSource	{swmf-att 13}	CHOICE	c28		c28		c28		-		-		c29		
20	identityOfCreator	{swmf-att 14}	GraphicString	c18		c19		c18		-		-		c18		
21	identityOfLastModifier	{swmf-att 15}	GraphicString	c18		c19		c18		-		-		c18		
22	lastBackupDestination	{swmf-att 16}	CHOICE	c30		c31		c30		-		-		c30		
23	lastBackupTime	{swmf-att 17}	CHOICE	c30		c31		c30		-		-		c30		
24	lastRestoreSource	{swmf-att 18}	CHOICE	c32		c33		c32		-		-		c32		
25	lastRestoreTime	{swmf-att 19}	CHOICE	c32		c33		c32		-		-		c32		
26	nameBinding	{dmi-att 63}	OBJECT IDENTIFIER	o		m		x		-		-		x		
27	noteField	{swmf-att 20}	GraphicString	c34		c34		c34		-		-		c35		
28	objectClass	{dmi-att 65}	CHOICE	m		m		x		-		-		x		

Table D.4 *(concluded)* – **Software Unit Attribute support**

				Set by create		Get		Replace		Add		Remove		Set to default		
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Addi-tional infor-mation
29	operationalState	{dmi-att 35}	ENUMERA-TED	x		m		x		-		-		x		
30	packages	{dmi-att 66}	SET OF OBJECT IDENTIFIER	o		m		x		x		x		x		
31	proceduralStatus	{dmi-att 36}	SET OF INTEGER	o		m		-		-		-		-		
32	softwareId	{m3100-att 38}	CHOICE	o		m		x		-		-		x		
33	usageState	{dmi-att 39}	ENUMERA-TED	-		o		-		-		-		-		
34	userLabel	{m3100-att 50}	GraphicString	o		o		o		-		-		-		
35	vendorName	{m3100-att 51}	GraphicString	o		o		o		-		-		-		
36	version	{m3100-att 52}	GraphicString	o		o		o		-		-		-		

c3: if D.4.1/1b then x else –.
c4: if D.4.3/2a and D.4.1/1b then x else –.
c4a: if D.4.3/2a then m else –.
c5: if D.4.3/21a and D.4.1/1b then x else –.
c5a: if D.4.3/21a then m else –.
c6: if D.4.3/3a then o else –.
c7: if D.4.3/3a then m else –.
c8: if D.4.3/4a and D.4.1/1b then x else –.
c9: if D.4.3/4a then m else –.
c10: if D.4.3/6a and D.4.1/1b then x else –.
c11: if D.4.3/6a then m else –.
c12: if D.4.3/8a and D.4.1/1b then x else –.
c13: if D.4.3/8a then m else –.
c14: if D.4.3/9a and D.4.1/1b then x else –.
c15: if D.4.3/9a then m else –.
c16: if D.4.3/9a and D.4.1/1b then x else –.
c17: if D.4.3/9a then m else –.
c18: if D.4.3/9a and D.4.1/1b then x else –.
c19: if D.4.3/9a then m else –.
c20: if D.4.3/9a and D.4.1/1b then x else –.
c21: if D.4.3/9a then m else –.
c22: if D.4.3/10a and D.4.1/1b then x else –.
c23: if D.4.3/10a then m else –.
c24: if D.4.3/10a and D.4.1/1b then x else –.
c25: if D.4.3/10a then m else –.
c26: if D.4.3/11a then m else –.
c27: if D.4.3/11a and D.4.1/1b then x else –.
c28: if D.4.3/12a then m else –.
c29: if D.4.3/12a and D.4.1/1b then x else –.
c30: if D.4.3/13a and D.4.1/1b then x else –.
c31: if D.4.3/13a then m else –.
c32: if D.4.3/14a and D.4.1/1b then x else –.
c33: if D.4.3/14a then m else –.
c34: if D.4.3/16a then m else –.
c35: if D.4.3/16a and D.4.1/1b then x else –.

D.4.4 Attribute groups

There are no attribute groups specified for this managed object class.

D.4.5 Actions

Table D.5 – Software Unit Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	backup	{swmf-act 7}	softwareProcessingFailureParameter	c1			1.1	BackupArgument	Information Syntax SEQUENCE	c1		
							1.1.1	backupDestination	CHOICE	c:m		
							1.1.1.1	localObject	Object Instance	c:o.1		
							1.1.1.2	inLine	NULL	c:o.1		
							1.1.1.3	offLine	GraphicString	c:o.1		
							1.1.2	additionalInfo	SET OF ANY	c:o		
							1.2	BackupReply	Reply Syntax SEQUENCE	c1		
							1.2.1	reply	CHOICE	c:m		
							1.2.1.1	success	NULL	c:o.2		
							1.2.1.2	inLine	BIT STRING	c:o.2		
							1.2.2	additionalInfo	SET OF ANY	c:o		
2	install	{swmf-act 3}	softwareProcessingFailureParameter	c2			2.1	InstallInfo	Information Syntax SEQUENCE	c2		
							2.1.1	targetSoftware	SET OF CHOICE	c:m		
							2.1.1.1	distributedSoftwareId	GraphicString	c:o.1		
							2.1.1.2	distributedSoftwarePointer	CHOICE	c:o.1		
							2.1.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.2		
							2.1.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							2.1.1.2.1.2	AttributeValue	ANY	c:m		
							2.1.1.2.2	nonSpecificForm	OCTET STRING	c:o.2		
							2.1.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.2		
							2.1.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							2.1.1.2.3.2	AttributeValue	ANY	c:m		
							2.1.2	installInfo	SET OF SEQUENCE	c:m		
							2.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							2.1.2.2	significance	BOOLEAN	c:o		
							2.1.2.3	information	ANY DEFINED BY identifier	c:m		

Table D.5 *(continued)* – **Software Unit Action support**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
3	restore	{swmf-act 8}		c3			3.1	RestoreArgument	Information Syntax SEQUENCE	c3		
							3.1.1	restoreSource	CHOICE	c:m		
							3.1.1.1	localObject	Object Instance	c:o.1		
							3.1.1.2	inLine	BIT STRING	c:o.1		
							3.1.1.3	offLine	Printable String	c:o.1		
							3.1.2	additionalInfo	SET OF ANY	c:o		
4	revert	{swmf-act 4}	softwareProcessingFailureParameter	c4			4.1	RevertArgument	Information Syntax SEQUENCE	c4		
							4.1.1	revertInfo	SEQUENCE OF CHOICE	c:m		
							4.1.1.1	patchId	GraphicString	c:o.3		
							4.1.1.2	patchPointer	CHOICE	c:o.3		
							4.1.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.4		
							4.1.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							4.1.1.2.1.2	AttributeValue	ANY	c:m		
							4.1.1.2.2	nonSpecificForm	OCTET STRING	c:o.4		
							4.1.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.4		
							4.1.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							4.1.1.2.3.2	AttributeValue	ANY	c:m		
							4.1.2	additionalInfo	SET OF SEQUENCE	c:o		
							4.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							4.1.2.2	significance	BOOLEAN	c:o		
							4.1.2.3	information	ANY DEFINED BY identifier	c:m		
							4.2	RevertReply	Reply Syntax SEQUENCE	c4		
							4.2.1	appliedPatches	SEQUENCE OF CHOICE	c:m		
							4.2.1.1	patchId	GraphicString	c:o.5		
							4.2.1.2	patchPointer	CHOICE	c:o.5		
							4.2.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.6		

Table D.5 *(continued)* – **Software Unit Action support**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							4.2.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							4.2.1.2.1.2	AttributeValue	ANY	c:m		
							4.2.1.2.2	nonSpecificForm	OCTET STRING	c:o.6		
							4.2.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.6		
							4.2.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							4.2.1.2.3.2	AttributeValue	ANY	c:m		
							4.2.2	additionalInfo	SET OF SEQUENCE	c:o		
							4.2.2.1	identifier	OBJECT IDENTIFIER	c:m		
							4.2.2.2	significance	BOOLEAN	c:o		
							4.2.2.3	information	ANY DEFINED BY identifier	c:m		
5	terminateValidation	{swmf-act 5}	softwareProcessingFailureParameter	c5			5.1	TerminateValidationArgument	Information Syntax SEQUENCE	c5		
							5.1.1	terminateValidationInfo	ENUMERATED	c:m		
							5.1.2	additionalInfo	SET OF SEQUENCE	c:o		
							5.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							5.1.2.2	significance	BOOLEAN	c:o		
							5.1.2.3	information	ANY DEFINED BY identifier	c:m		
							5.2	TerminateValidationReply	Reply Syntax CHOICE	c5		
							5.2.1	noOutStandingValidation	NULL	c:o.7		
							5.2.2	validationCancelled	NULL	c:o.7		
							5.2.3	resultOfPartialValidation	CHOICE	c:o.7		
							5.2.3.1	validationTerminated	NULL	c:o.8		
							5.2.3.2	passValidation	NULL	c:o.8		
							5.2.3.3	passValidationWithResult	SET OF SEQUENCE	c:o.8		
							5.2.3.3.1	identifier	OBJECT IDENTIFIER	c:m		
							5.2.3.3.2	significance	BOOLEAN	c:o		
							5.2.3.3.3	information	ANY DEFINED BY identifier	c:m		

Table D.5 *(concluded)* – **Software Unit Action support**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							5.2.3.4	failValidation	NULL	c:o.8		
							5.2.3.5	failValidationWithResult	SET OF SEQUENCE	c:o.8		
							5.2.3.5.1	identifier	OBJECT IDENTIFIER	c:m		
							5.2.3.5.2	significance	BOOLEAN	c:o		
							5.2.3.5.3	information	ANY DEFINED BY identifier	c:m		
6	validate	{swmf-act 6}	softwareProcessingFailureParameter	c6			9.1	ValidateInfo	Information Syntax CHOICE	c6		
							9.1.1	instanceDefaultValidationType	NULL	c:o.9		
							9.1.2	registeredValidationType	OBJECT IDENTIFIER	c:o.9		
							9.1.3	systemSpecificValidationType	SET OF SEQUENCE	c:o.9		
							9.1.3.1	identifier	OBJECT IDENTIFIER	c:m		
							9.1.3.2	significance	BOOLEAN	c:o		
							9.1.3.3	information	ANY DEFINED BY identifier	c:m		
							9.2	ValidateReply	Reply Syntax CHOICE	c6		
							9.2.1	validationTerminated	NULL	c:o.10		
							9.2.2	passValidation	NULL	c:o.10		
							9.2.3	passValidationWithResult	SET OF SEQUENCE	c:o.10		
							9.2.3.1	identifier	OBJECT IDENTIFIER	c:m		
							9.2.3.2	significance	BOOLEAN	c:o		
							9.2.3.3	information	ANY DEFINED BY identifier	c:m		
							9.2.4	failValidation	NULL	c:o.10		
							9.2.5	failValidationWithResult	SET OF SEQUENCE	c:o.10		
							9.2.5.1	identifier	OBJECT IDENTIFIER	c:m		
							9.2.5.2	significance	BOOLEAN	c:o		
							9.2.5.3	information	ANY DEFINED BY identifier	c:m		

c1: if D.4.3/13a then m else –.
c2: if D.4.2/15a then m else –.
c3: if D.4.2/14a then m else –.
c4: if D.4.2/19a then m else –.
c5: if D.4.2/24a then m else –.
c6: if D.4.2/28a then m else –.

D.4.6 Notifications

Table D.6 – Software Unit Notification support

					Support									
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	attributeValueChange	{dmi-not1}		c1				1.1	AttributeValueChangeInfo		Information Syntax SEQUENCE	c1		
								1.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								1.1.2	attributeIdentifierList	{dmi-att 8}	SET OF CHOICE	c:o		
								1.1.2.1	globalForm	-	OBJECT IDENTIFIER	c:o.1		
								1.1.2.2	localForm	-	INTEGER	c:o.1		
								1.1.3	attributeValueChangeDefinition	{dmi-att 10}	SET OF SEQUENCE	c:m		
								1.1.3.1	attributeID	-	CHOICE	c:m		
								1.1.3.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.2		
								1.1.3.1.2	localForm	-	INTEGER	c:o.2		
								1.1.3.2	oldAttributeValue	-	ANY DEFINED BY attributeID	c:o		
								1.1.3.3	newAttributeValue	-	ANY DEFINED BY attributeID	c:m		
								1.1.4	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								1.1.5	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								1.1.5.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								1.1.5.2	sourceObjectInst	-	CHOICE	c:o		
								1.1.5.2.1	distinguishedNamc	-	SEQUENCE OF SET OF SEQUENCE	c:o.3		
								1.1.5.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								1.1.5.2.1.2	AttributeValue	-	ANY	c:m		
								1.1.5.2.2	nonSpecificForm	-	OCTET STRING	c:o.3		
								1.1.5.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.3		
								1.1.5.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								1.1.5.2.3.2	AttributeValuc	-	ANY	c:m		
								1.1.6	additionalText	{dmi-att 7}	GraphicString	c:o		
								1.1.7	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								1.1.7.1	identifier	-	OBJECT IDENTIFIER	c:m		

Table D.6 *(continued)* **– Software Unit Notification support**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								1.1.7.2	significance	-	BOOLEAN	c:o		
								1.1.7.3	information	-	ANY DEFINED BY identifier	c:m		
2	autoBackupReport	{2 9 2 18 10 1}		c2				2.1	AutoBackupReportInfo		Information Syntax SEQUENCE	c2		
								2.1.1	backupResult	-	CHOICE	c:m		
								2.1.1.1	inLine	-	CHOICE	c:o.4		
								2.1.1.2	local	-	SEQUENCE	c:o.4		
								2.1.1.2.1	destination	-	CHOICE	c:m		
								2.1.1.2.1.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.5		
								2.1.1.2.1.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								2.1.1.2.1.1.2	AttributeValue	-	ANY	c:m		
								2.1.1.2.1.2	nonSpecificForm	-	OCTET STRING	c:o.5		
								2.1.1.2.1.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.5		
								2.1.1.2.1.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								2.1.1.2.1.3.2	AttributeValue	-	ANY	c:m		
								2.1.1.2.2	success	-	BOOLEAN	c:m		
								2.1.1.3	offLine	-	SEQUENCE	c:o.4		
								2.1.1.3.1	destination	-	GraphicString	c:m		
								2.1.1.3.2	result	-	CHOICE	c:m		
3	autoRestoreReport	{2 9 2 18 10 2}		c3				3.1	AutoRestoreReportInfo		Information Syntax SEQUENCE	c3		
								3.1.1	source	-	CHOICE	c:m		
								3.1.1.1	localObject	-	CHOICE	c:o.6		
								3.1.1.1.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.7		
								3.1.1.1.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								3.1.1.1.1.2	AttributeValue	-	ANY	c:m		
								3.1.1.1.2	nonSpecificForm	-	OCTET STRING	c:o.7		
								3.1.1.1.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.7		
								3.1.1.1.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								3.1.1.1.3.2	AttributeValue	-	ANY	c:m		

Table D.6 *(continued)* – **Software Unit Notification support**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								3.1.1.2	remoteSystem	-	GraphicString	c:o.6		
								3.1.2	success	-	BOOLEAN	c:m		
								3.1.3	additionalInfo	-	SET OF SEQUENCE	c:o		
								3.1.3.1	identifier	-	OBJECT IDENTIFIER	c:m		
								3.1.3.2	significance	-	BOOLEAN	c:o		
								3.1.3.3	information	-	ANY DEFINED BY identifier	c:m		
4	objectCreation	{dmi-not6}		c4				4.1	ObjectInfo		Information Syntax SEQUENCE	c4		
								4.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								4.1.2	attributeList	{dmi-att 9}	SET OF SEQUENCE	c:o		
								4.1.2.1	attributeId	-	CHOICE	c:m		
								4.1.2.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.8		
								4.1.2.1.2	localForm	-	INTEGER	c:o.8		
								4.1.2.2	attributeValue	-	ANY DEFINED BY attributeId	c:m		
								4.1.3	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								4.1.4	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								4.1.4.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								4.1.4.2	sourceObjectInst	-	CHOICE	c:o		
								4.1.4.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.9		
								4.1.4.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								4.1.4.2.1.2	AttributeValue	-	ANY	c:m		
								4.1.4.2.2	nonSpecificForm	-	OCTET STRING	c:o.9		
								4.1.4.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.9		
								4.1.4.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								4.1.4.2.3.2	AttributeValue	-	ANY	c:m		
								4.1.5	additionalText	{dmi-att 7}	GraphicString	c:o		
								4.1.6	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								4.1.6.1	identifier	-	OBJECT IDENTIFIER	c:m		

Table D.6 *(continued)* – **Software Unit Notification support**

					Support										
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								4.1.6.2	significance	-	BOOLEAN	c:o		
								4.1.6.3	information	-	ANY DEFINED BY identifier	c:m		
5	objectDeletion	{dmi-not7}		e5				5.1	ObjectInfo		Information Syntax SEQUENCE	e5		
								5.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								5.1.2	attributeList	{dmi-att 9}	SET OF SEQUENCE	c:o		
								5.1.2.1	attributeId	-	CHOICE	c:m		
								5.1.2.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 10		
								5.1.2.1.2	localForm	-	INTEGER	c:o. 10		
								5.1.2.2	attributeValue	-	ANY DEFINED BY attributeId	c:m		
								5.1.3	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								5.1.4	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								5.1.4.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								5.1.4.2	sourceObjectInst	-	CHOICE	c:o		
								5.1.4.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11		
								5.1.4.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								5.1.4.2.1.2	AttributeValue	-	ANY	c:m		
								5.1.4.2.2	nonSpecificForm	-	OCTET STRING	c:o. 11		
								5.1.4.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11		
								5.1.4.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								5.1.4.2.3.2	AttributeValue	-	ANY	c:m		
								5.1.5	additionalText	{dmi-att 7}	GraphicString	c:o		
								5.1.6	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								5.1.6.1	identifier	-	OBJECT IDENTIFIER	c:m		
								5.1.6.2	significance	-	BOOLEAN	c:o		
								5.1.6.3	information	-	ANY DEFINED BY identifier	c:m		

Table D.6 *(continued)* **– Software Unit Notification support**

					Support										
Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Confirmed	Non-confirmed	Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
6	processingErrorAlarm	{dmi-not10}		m			alarmEffectOnServiceParameter	6.1	AlarmInfo		Information Syntax SEQUENCE	m		
								6.1.1	probableCause	{dmi-att 18}	CHOICE	m		
								6.1.1.1	globalValue	-	OBJECT IDENTIFIER	o.12		
								6.1.1.2	localValue	-	INTEGER	o.12		
								6.1.2	specificProblems	{dmi-att 27}	SET OF CHOICE	o		
								6.1.2.1	OBJECT IDENTIFIER	-	OBJECT IDENTIFIER	c:o.13		
								6.1.2.2	INTEGER	-	INTEGER	c:o.13		
								6.1.3	perceivedSeverity	{dmi-att 17}	ENUMERATED	m		
								6.1.4	backedUpStatus	{dmi-att 11}	BOOLEAN	o		
								6.1.5	backUpObject	{dmi-att 40}	CHOICE	o		
								6.1.5.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.14		
								6.1.5.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.5.1.2	AttributeValue	-	ANY	c:m		
								6.1.5.2	nonSpecificForm	-	OCTET STRING	c:o.14		
								6.1.5.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.14		
								6.1.5.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.5.3.2	AttributeValue	-	ANY	c:m		
								6.1.6	trendIndication	{dmi-att 30}	ENUMERATED	o		
								6.1.7	thresholdInfo	{dmi-att 29}	SEQUENCE	o		
								6.1.7.1	triggeredThreshold	-	CHOICE	c:m		
								6.1.7.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.15		
								6.1.7.1.2	localForm	-	INTEGER	c:o.15		
								6.1.7.2	observedValue	-	CHOICE	c:m		
								6.1.7.2.1	integer	-	INTEGER	c:o.16		

Table D.6 *(continued)* – **Software Unit Notification support**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								6.1.7.2.2	real	-	REAL	c:o. 16		
								6.1.7.3	thresholdLevel	-	CHOICE	c:o		
								6.1.7.3.1	up	-	SEQUENCE	c:o. 17		
								6.1.7.3.1.1	high	-	CHOICE	c:m		
								6.1.7.3.1.1.1	integer	-	INTEGER	c:o. 18		
								6.1.7.3.1.1.2	real	-	REAL	c:o. 18		
								6.1.7.3.1.2	low	-	CHOICE	c:o		
								6.1.7.3.1.2.1	integer	-	INTEGER	c:o. 19		
								6.1.7.3.1.2.2	real	-	REAL	c:o. 19		
								6.1.7.3.2	down	-	SEQUENCE	c:o. 17		
								6.1.7.3.2.1	high	-	CHOICE	c:m		
								6.1.7.3.2.1.1	integer	-	INTEGER	c:o. 20		
								6.1.7.3.2.1.2	real	-	REAL	c:o. 20		
								6.1.7.3.2.2	low	-	CHOICE	c:m		
								6.1.7.3.2.2.1	integer	-	INTEGER	c:o. 21		
								6.1.7.3.2.2.2	real	-	REAL	c:o. 21		
								6.1.7.4	armTime	-	GeneralizedTime	c:o		
								6.1.8	notificationIdentifier	{dmi-att 16}	INTEGER	o		
								6.1.9	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	o		
								6.1.9.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								6.1.9.2	sourceObjectInst	-	CHOICE	c:o		
								6.1.9.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 22		
								6.1.9.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.9.2.1.2	AttributeValue	-	ANY	c:m		
								6.1.9.2.2	nonSpecificForm	-	OCTET STRING	c:o. 22		
								6.1.9.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 22		

Table D.6 *(continued)* **– Software Unit Notification support**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								6.1.9.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								6.1.9.2.3.2	AttributeValue	-	ANY	c:m		
								6.1.10	stateChangeDefinition	{dmi-att 28}	SET OF SEQUENCE	o		
								6.1.10.1	attributeID	-	CHOICE	c:m		
								6.1.10.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 23		
								6.1.10.1.2	localForm	-	INTEGER	c:o. 23		
								6.1.10.2	oldAttributeValue	-	ANY DEFINED BY attributeID	c:o		
								6.1.10.3	newAttributeValue	-	ANY DEFINED BY attributeID	c:m		
								6.1.11	monitoredAttributes	{dmi-att 15}	SET OF SEQUENCE	o		
								6.1.11.1	attributeId	-	CHOICE	c:m		
								6.1.11.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 24		
								6.1.11.1.2	localForm	-	INTEGER	c:o. 24		
								6.1.11.2	attributeValue	-	ANY DEFINED BY attributeId	c:m		
								6.1.12	proposedRepairActions	{dmi-att 19}	SET OF CHOICE	o		
								6.1.12.1	OBJECT IDENTIFIER	-	OBJECT IDENTIFIER	c:o. 25		
								6.1.12.2	INTEGER	-	INTEGER	c:o. 25		
								6.1.13	additionalText	{dmi-att 7}	GraphicString	o		
								6.1.14	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	o		
								6.1.14.1	identifier	-	OBJECT IDENTIFIER	c:m		
								6.1.14.2	significance	-	BOOLEAN	c:o		
								6.1.14.3	information	-	ANY DEFINED BY identifier	c:m		
7	stateChange	{dmi-not14}		c6				7.1	StateChangeInfo		Information Syntax SEQUENCE	c6		
								7.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
								7.1.2	attributeIdentifierList	{dmi-att 8}	SET OF CHOICE	c:o		
								7.1.2.1	globalForm	-	OBJECT IDENTIFIER	c:o. 26		
								7.1.2.2	localForm	-	INTEGER	c:o. 26		

Table D.6 *(concluded)* – **Software Unit Notification support**

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support: Confirmed	Support: Non-confirmed	Support: Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
								7.1.3	stateChangeDefinition	{dmi-att 28}	SET OF SEQUENCE	c:m		
								7.1.3.1	attributeID	-	CHOICE	c:m		
								7.1.3.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 27		
								7.1.3.1.2	localForm	-	INTEGER	c:o. 27		
								7.1.3.2	oldAttributeValue	-	ANY DEFINED BY attributeID	c:o		
								7.1.3.3	newAttributeValue	-	ANY DEFINED BY attributeID	c:m		
								7.1.4	notificationIdentifier	{dmi-att 16}	INTEGER	c:o		
								7.1.5	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								7.1.5.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m		
								7.1.5.2	sourceObjectInst	-	CHOICE	c:o		
								7.1.5.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 28		
								7.1.5.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								7.1.5.2.1.2	AttributeValue	-	ANY	c:m		
								7.1.5.2.2	nonSpecificForm	-	OCTET STRING	c:o. 28		
								7.1.5.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 28		
								7.1.5.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
								7.1.5.2.3.2	AttributeValue	-	ANY	c:m		
								7.1.6	additionalText	{dmi-att 7}	GraphicString	c:o		
								7.1.7	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								7.1.7.1	identifier	-	OBJECT IDENTIFIER	c:m		
								7.1.7.2	significance	-	BOOLEAN	c:o		
								7.1.7.3	information	-	ANY DEFINED BY identifier	c:m		

c1: if D.4.3/5a then m else –.
c2: if D.4.3/11a then m else –.
c3: if D.4.3/12a then m else –.
c4: if D.4.3/7a then m else –.
c5: if D.4.3/7a then m else –.
c6: if D.4.3/23a then m else –.

D.4.7 Parameters

Table D.7 – Software Unit Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	alarmEffectOnServiceParameter	{2 9 2 18 5 1}	EVENT-INFO processingErrorAlarm	m		
2	softwareProcessingFailureParameter	{2 9 2 18 5 2}	SPECIFIC-ERROR	c1		
c1: if D.4.3/13a or D.4.3/15a or D.4.3/19a or D.4.3/24a or D.4.3/28a then m else –.						

D.5 Executable software managed object class

D.5.1 Statement of conformance to the managed object class

The supplier of the implementation shall state whether or not all mandatory features of the executable software managed object class are supported, and if the actual class supported is the same as the managed object class to which conformance is claimed.

Table D.8 – Executable Software Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	executableSoftware	{swmf-obj 2}		

If the answer to the actual class question in Table D.8 is "N", the supplier of the implementation shall supply the actual class support details, in Table D.10.

Table D.9 – Executable Software Actual class support

Index	Managed object class template for actual class	Value of object identifier for managed object class definition of actual class	Additional information

D.5.2 Packages

Table D.10 — Executable Software Package support

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	administrativeOperationalStatesPackage	{m3100-pkg 1}	Mandatory	m		
2	affectedObjectListPackage	{m3100-pkg 2}	"an instance supports it"	o		
3	allomorphicPackage	{dmi-pkg 17}	"if an object supports allomorphism"	c1		
4	appliedPatchPackage	{swmf-pkg 1}	"an instance supports software patching"	o		
5	attributeValueChangeNotificationPackage	{m3100-pkg 4}	"the attributeValueChange notification defined in Recommendation X.721 is supported by an instance of this class."	o		
6	checkSumPackage	{swmf-pkg 2}	"an instance supports it"	o		
7	createDeleteNotificationsPackage	{m3100-pkg 10}	"the objectCreation and objectDeletion notifications defined in Recommendation X.721 are supported by an instance of this class."	o		
8	currentProblemListPackage	{m3100-pkg 13}	"an instance supports it"	o		
9	fileInformationPackage	{swmf-pkg 4}	"an instance supports it"	o		
10	filePackage	{swmf-pkg 5}	"an instance supports it"	o		
11	informationAutoBackupPackage	{swmf-pkg 6}	"an instance supports it"	o		
12	informationAutoRestorePackage	{swmf-pkg 7}	"an instance supports it"	o		
13	informationBackupPackage	{swmf-pkg 8}	"an instance supports it"	o		
14	informationRestorePackage	{swmf-pkg 9}	"an instance supports it"	o		
15	installPackage	{swmf-pkg 10}	"an instance supports it"	o		
16	noteFieldPackage	{swmf-pkg 11}	"an instance supports it"	o		
17	packagesPackage	{dmi-pkg 16}	"any registered package, other than this package has been instantiated"	m		
18	processingErrorAlarmOnServicePackage	{swmf-pkg 12}	Mandatory	m		
19	revertPackage	{swmf-pkg 13}	"an instance supports it"	o		
20	softwarePackage		Mandatory	m		
21	softwareProcessingErrorAlarmPackage	{m3100-pkg 26}	"an instance supports it"	o		
22	softwareUnitPackage		Mandatory	m		
23	stateChangeNotificationPackage	{m3100-pkg 28}	"the stateChange notification defined in Recommendation X.721 is supported by an instance of this class."	o		

Table D.10 *(concluded)* **– Executable Software Package support**

Index	Parameter template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
24	terminateValidationPackage	{swmf-pkg 14}	"the validationPackage is present and an instance supports it"	c2		
25	topPackage		Mandatory	m		
26	usageStatePackage	{swmf-pkg 15}	"an instance supports it"	o		
27	userLabelPackage	{m3100-pkg 32}	"an instance supports it"	o		
28	validationPackage	{swmf-pkg 16}	"an instance supports it"	o		
29	vendorNamePackage	{m3100-pkg 33}	"an instance supports it"	o		
30	versionPackage	{m3100-pkg 34}	"an instance supports it"	o		
31	executableSoftwarePackage	(Not registered)	Mandatory	m		
32	executeProgramPackage	{swmf-pkg 3}	"an instance supports it"	o		
c1: if D.5.1/1b then – else m. c2: if D.5.3/28a then o else –.						

D.5.3 Attributes

Table D.11 – Executable Software Attribute support

				Set by create		Get		Replace		Add		Remove		Set to default		
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	administrativeState	{dmi-att 31}	ENUMERATED	m		m		m		-		-		c3		
2	affectedObjectList	{m3100-att 2}	SET OF CHOICE	c4		c4a		c4		c4		c4		c4		
3	alarmStatus	{m3100-att 6}	ENUMERATED	c5		c5a		c5		-		-		c5		
4	allomorphs	{dmi-att 50}	SET OF CHOICE	c6		c7		-		-		-		-		
5	appliedPatches	{swmf-att 1}	SEQUENCE OF CHOICE	c8		c9		c8		-		-		c8		
6	availabilityStatus	{dmi-att 33}	SET OF INTEGER	c3		m		c3		c3		c3		c3		
7	checkSum	{swmf-att 2}	BIT STRING	c10		c11		c10		-		-		c10		
8	currentProblemList	{m3100-att 17}	SET OF SEQUENCE	c12		c13		c12		c12		c12		c12		

Table D.11 *(continued)* – **Executable Software Attribute support**

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get		Replace		Add		Remove		Set to default		Additional information
				Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	
9	dateDelivered	{swmf-att 3}	CHOICE	c14		c15		c14		-		-		c14		
10	dateInstalled	{swmf-att 4}	CHOICE	c16		c17		c16		-		-		c16		
11	dateOfCreation	{swmf-att 5}	GeneralizedTime	c18		c19		c18		-		-		c18		
12	dateOfLastModification	{swmf-att 6}	CHOICE	c20		c21		c20		-		-		c20		
13	fileLocation	{swmf-att 7}	CHOICE	c22		c23		c22		-		-		c22		
14	fileSize	{swmf-att 8}	CHOICE	c24		c25		c24		-		-		c24		
15	filetType	{swmf-att 9}		o		o		-		-		-		-		
16	futureAutoBackupDestination	{swmf-att 10}	BackupDestination	c26		c26		c26		-		-		c27		
17	futureAutoBackupTriggerThreshold	{swmf-att 11}	INTEGER	c26		c26		c26		-		-		c27		
18	futureAutoRestoreAllowed	{swmf-att 12}	BOOLEAN	c28		c28		c28		-		-		c29		
19	futureAutoRestoreSource	{swmf-att 13}	CHOICE	c28		c28		c28		-		-		c29		
20	identityOfCreator	{swmf-att 14}	GraphicString	c18		c19		c18		-		-		c18		
21	identityOfLastModifier	{swmf-att 15}	GraphicString	c18		c19		c18		-		-		c18		
22	lastBackupDestination	{swmf-att 16}	CHOICE	c30		c31		c30		-		-		c30		
23	lastBackupTime	{swmf-att 17}	CHOICE	c30		c31		c30		-		-		c30		
24	lastRestoreSource	{swmf-att 18}	CHOICE	c32		c33		c32		-		-		c32		
25	lastRestoreTime	{swmf-att 19}	CHOICE	c32		c33		c32		-		-		c32		
26	nameBinding	{dmi-att 63}	OBJECT IDENTIFIER	o		m		x		-		-		x		
27	noteField	{swmf-att 20}	GraphicString	c34		c34		c34		-		-		c35		
28	objectClass	{dmi-att 65}	CHOICE	m		m		x		-		-		x		
29	operationalState	{dmi-att 35}	ENUMERATED	x		m		x		-		-		x		
30	packages	{dmi-att 66}	SET OF OBJECT IDENTIFIER	o		m		x		x		x		x		

Table D.11 *(concluded)* – **Executable Software Attribute support**

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get		Replace		Add		Remove		Set to default		Addi-tional infor-mation
				Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	Sta-tus	Sup-port	
31	proceduralStatus	{dmi-att 36}	SET OF INTEGER	o		m		-		-		-		-		
32	softwareId	{m3100-att 38}	CHOICE	o		m		x		-		-		x		
33	usageState	{dmi-att 39}	ENUMERATED	c36		m		c36		-		-		c36		
34	userLabel	{m3100-att 50}	GraphicString	o		o		o		-		-		-		
35	vendorName	{m3100-att 51}	GraphicString	o		o		o		-		-		-		
36	version	{m3100-att 52}	GraphicString	o		o		o		-		-		-		

c3: if D.5.1/1b then x else –.
c4: if D.5.3/2a and D.5.1/1b then x else –.
c4a: if D.5.3/2a then m else –.
c5: if D.5.3/21a and D.5.1/1b then x else –.
c5a: if D.5.3/21a then m else –.
c6: if D.5.3/3a then o else –.
c7: if D.5.3/3a then m else –.
c8: if D.5.3/4a and D.5.1/1b then x else –.
c9: if D.5.3/4a then m else –.
c10: if D.5.3/6a and D.5.1/1b then x else –.
c11: if D.5.3/6a then m else –.
c12: if D.5.3/8a and D.5.1/1b then x else –.
c13: if D.5.3/8a then m else –.
c14: if D.5.3/9a and D.5.1/1b then x else –.
c15: if D.5.3/9a then m else –.
c16: if D.5.3/9a and D.5.1/1b then x else –.
c17: if D.5.3/9a then m else –.
c18: if D.5.3/9a and D.5.1/1b then x else –.
c19: if D.5.3/9a then m else –.
c20: if D.5.3/9a and D.5.1/1b then x else –.
c21: if D.5.3/9a then m else –.
c22: if D.5.3/10a and D.5.1/1b then x else –.
c23: if D.5.3/10a then m else –.
c24: if D.5.3/10a and D.5.1/1b then x else –.
c25: if D.5.3/10a then m else –.
c26: if D.5.3/11a then m else –.
c27: if D.5.3/11a and D.5.1/1b then x else –.
c28: if D.5.3/12a then m else –.
c29: if D.5.3/12a and D.5.1/1b then x else –.
c30: if D.5.3/13a and D.5.1/1b then x else –.
c31: if D.5.3/13a then m else –.
c32: if D.5.3/14a and D.5.1/1b then x else –.
c33: if D.5.3/14a then m else –.
c34: if D.5.3/16a then m else –.
c35: if D.5.3/16a and D.5.1/1b then x else –.
c36: if D.5.1/1b then x else –.

D.5.4 Attribute groups

There are no attribute groups specified for this managed object class.

D.5.5 Actions

Table D.12 – Executable Software Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	backup	{swmf-act 7}	softwareProcessingFailureParameter	c1			1.1	BackupArgument	Information Syntax SEQUENCE	c1		
							1.1.1	restoreSource	CHOICE	c:m		
							1.1.1.1	localObject	Object Instance	c:o.1		
							1.1.1.2	inLine	BIT STRING	c:o.1		
							1.1.1.3	offLine	GraphicString	c:o.1		
							1.1.2	additionalInfo	SET OF ANY	c:o		
							1.2	BackupReply	Reply Syntax SEQUENCE	c1		
							1.2.1	reply	CHOICE	c:m		
							1.2.1.1	success	NULL	c:o.2		
							1.2.1.2	inLine	BIT STRING	c:o.2		
							1.2.2	additionalInfo	SET OF ANY	c:o		
2	executeProgram	{swmf-act 2}	softwareProcessingFailureParameter	c7			2.1	executeProgramInfo	Information Syntax SET OF SEQUENCE	c7		
							2.1.2	significance	BOOLEAN	c:o		
							2.1.3	information	ANY DEFINED BY identifier	c:m		
							2.2	ExecuteProgramReply	Reply Syntax SEQUENCE	c7		
							2.2.1	processId	INTEGER	c:m		
							2.2.2	processOwner	GraphicString	c:m		
							2.2.3	startTime	GeneralizedTime	c:m		
							2.2.4	additionalInfo	SET OF SEQUENCE	c:o		
							2.2.4.1	identifier	OBJECT IDENTIFIER	c:m		
							2.2.4.2	significance	BOOLEAN	c:o		
							2.2.4.3	information	ANY DEFINED BY identifier	c:m		
3	install	{swmf-act 3}	softwareProcessingFailureParameter	c2			3.1	InstallInfo	Information Syntax SEQUENCE	c2		
							3.1.1	targetSoftware	SET OF CHOICE	c:m		
							3.1.1.1	distibutedSoftwareId	GraphicString	c:o.1		

Table D.12 *(continued)* – **Executable Software Action support**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Sta-tus	Sup-port	Addi-tional infor-mation	Subindex	Action field name label	Constraints and values	Status	Sup-port	Addi-tional infor-mation
							3.1.1.2	distributedSoftwarePointer	CHOICE	c:o.1		
							3.1.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.2		
							3.1.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							3.1.1.2.1.2	AttributeValue	ANY	c:m		
							3.1.1.2.2	nonSpecificForm	OCTET STRING	c:o.2		
							3.1.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.2		
							3.1.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							3.1.1.2.3.2	AttributeValue	ANY	c:m		
							3.1.2	installInfo	SET OF SEQUENCE	c:m		
							3.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							3.1.2.2	significance	BOOLEAN	c:o		
							3.1.2.3	information	ANY DEFINED BY identifier	c:m		
4	restore	{swmf-act 8}		c3			4.1	RestoreArgument	Information Syntax SEQUENCE	c3		
							4.1.1	restoreSource	CHOICE	c:m		
							4.1.1.1	localObject	Object Instance	c:o.1		
							4.1.1.2	inLine	BIT STRING	c:o.1		
							4.1.1.3	offLine	Printable String	c:o.1		
							4.1.2	additionalInfo	SET OF ANY	c:o		
5	revert	{swmf-act 4}	softwareProcessingFailureParameter	c4			5.1	RevertArgument	Information Syntax SEQUENCE	c4		
							5.1.1	revertInfo	SEQUENCE OF CHOICE	c:m		
							5.1.1.1	patchId	GraphicString	c:o.3		
							5.1.1.2	patchPointer	CHOICE	c:o.3		
							5.1.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.4		
							5.1.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		

Table D.12 *(continued)* – **Executable Software Action support**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Sta-tus	Sup-port	Addi-tional infor-mation	Subindex	Action field name label	Constraints and values	Status	Sup-port	Addi-tional infor-mation
							5.1.1.2.1.2	AttributeValue	ANY	c:m		
							5.1.1.2.2	nonSpecificForm	OCTET STRING	c:o.4		
							5.1.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.4		
							5.1.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							5.1.1.2.3.2	AttributeValue	ANY	c:m		
							5.1.2	additionalInfo	SET OF SEQUENCE	c:o		
							5.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							5.1.2.2	significance	BOOLEAN	c:o		
							5.1.2.3	information	ANY DEFINED BY identifier	c:m		
							5.2	RevertReply	Reply Syntax SEQUENCE	c4		
							5.2.1	appliedPatches	SEQUENCE OF CHOICE	c:m		
							5.2.1.1	patchId	GraphicString	c:o.5		
							5.2.1.2	patchPointer	CHOICE	c:o.5		
							5.2.1.2.1	distinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.6		
							5.2.1.2.1.1	AttributeType	OBJECT IDENTIFIER	c:m		
							5.2.1.2.1.2	AttributeValue	ANY	c:m		
							5.2.1.2.2	nonSpecificForm	OCTET STRING	c:o.6		
							5.2.1.2.3	localDistinguishedName	SEQUENCE OF SET OF SEQUENCE	c:o.6		
							5.2.1.2.3.1	AttributeType	OBJECT IDENTIFIER	c:m		
							5.2.1.2.3.2	AttributeValue	ANY	c:m		
							5.2.2	additionalInfo	SET OF SEQUENCE	c:o		
							5.2.2.1	identifier	OBJECT IDENTIFIER	c:m		
							5.2.2.2	significance	BOOLEAN	c:o		
							5.2.2.3	information	ANY DEFINED BY identifier	c:m		

Table D.12 *(continued)* – **Executable Software Action support**

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
6	terminateValidation	{swmf-act 5}	softwareProcessingFailureParameter	c5			6.1	TerminateValidationArgument	Information Syntax SEQUENCE	c5		
							6.1.1	terminateValidationInfo	ENUMERATED	c:m		
							6.1.2	additionalInfo	SET OF SEQUENCE	c:o		
							6.1.2.1	identifier	OBJECT IDENTIFIER	c:m		
							6.1.2.2	significance	BOOLEAN	c:o		
							6.1.2.3	information	ANY DEFINED BY identifier	c:m		
							6.2	TerminateValidationReply	Reply Syntax CHOICE	c5		
							6.2.1	noOutStandingValidation	NULL	c:o.7		
							6.2.2	validationCancelled	NULL	c:o.7		
							6.2.3	resultOfPartialValidation	CHOICE	c:o.7		
							6.2.3.1	validationTerminated	NULL	c:o.8		
							6.2.3.2	passValidation	NULL	c:o.8		
							6.2.3.3	passValidationWithResult	SET OF SEQUENCE	c:o.8		
							6.2.3.3.1	identifier	OBJECT IDENTIFIER	c:m		
							6.2.3.3.2	significance	BOOLEAN	c:o		
							6.2.3.3.3	information	ANY DEFINED BY identifier	c:m		
							6.2.3.4	failValidation	NULL	c:o.8		
							6.2.3.5	failValidationWithResult	SET OF SEQUENCE	c:o.8		
							6.2.3.5.1	identifier	OBJECT IDENTIFIER	c:m		
							6.2.3.5.2	significance	BOOLEAN	c:o		
							6.2.3.5.3	information	ANY DEFINED BY identifier	c:m		