



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.744

(10/96)

SÉRIE X: RÉSEAUX POUR DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS
Gestion OSI

**Technologies de l'information –
Interconnexion des systèmes ouverts –
Gestion-systèmes: fonction de gestion de
logiciel**

Recommandation UIT-T X.744

(Antérieurement «Recommandation du CCITT»)

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX DE DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS POUR DONNÉES	X.1-X.199
Services et fonctionnalités	X.1-X.19
Interfaces	X.20-X.49
Transmission, signalisation et commutation	X.50-X.89
Aspects réseau	X.90-X.149
Maintenance	X.150-X.179
Dispositions administratives	X.180-X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	X.200-X.299
Modèle et notation	X.200-X.209
Définitions des services	X.210-X.219
Spécifications des protocoles en mode connexion	X.220-X.229
Spécifications des protocoles en mode sans connexion	X.230-X.239
Formulaires PICS	X.240-X.259
Identification des protocoles	X.260-X.269
Protocoles de sécurité	X.270-X.279
Objets gérés de couche	X.280-X.289
Tests de conformité	X.290-X.299
INTERFONCTIONNEMENT DES RÉSEAUX	X.300-X.399
Généralités	X.300-X.349
Systèmes de transmission de données par satellite	X.350-X.399
SYSTÈMES DE MESSAGERIE	X.400-X.499
ANNUAIRE	X.500-X.599
RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES	X.600-X.699
Réseautage	X.600-X.629
Efficacité	X.630-X.649
Dénomination, adressage et enregistrement	X.650-X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680-X.699
GESTION OSI	X.700-X.799
Cadre général et architecture de la gestion-systèmes	X.700-X.709
Service et protocole de communication de gestion	X.710-X.719
Structure de l'information de gestion	X.720-X.729
Fonctions de gestion	X.730-X.799
SÉCURITÉ	X.800-X.849
APPLICATIONS OSI	X.850-X.899
Engagement, concomitance et rétablissement	X.850-X.859
Traitement transactionnel	X.860-X.879
Opérations distantes	X.880-X.899
TRAITEMENT OUVERT RÉPARTI	X.900-X.999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI. Le texte de la Recommandation X.744 de l'UIT-T a été approuvé par la CMNT (Genève, 9-18 octobre 1996). Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10164-18.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

© UIT 1997

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT, sauf en ce qui concerne les Notes de bas de page 3) à 7) des Annexes B à F respectivement.

TABLE DES MATIÈRES

	<i>Page</i>
1 Domaine d'application.....	1
2 Références normatives	2
2.1 Recommandations Normes internationales identiques.....	2
2.2 Paires de Recommandations Normes internationales équivalentes par leur contenu technique	3
2.3 Références additionnelles	3
3 Définitions.....	3
3.1 Définitions du cadre général de gestion	4
3.2 Définitions de l'aperçu général de la gestion-systèmes.....	4
3.3 Définitions du service CMIS.....	4
3.4 Définitions du modèle d'information de gestion	4
3.5 Directives pour l'établissement de la définition des objets gérés	4
3.6 Définitions des formulaires de déclaration de conformité d'implémentation.....	4
3.7 Définitions supplémentaires.....	4
4 Symboles et abréviations.....	5
5 Conventions.....	6
6 Prescriptions pour la gestion de logiciel.....	6
7 Modèle concernant la fonction de gestion de logiciel.....	6
7.1 Fonctions de gestion de logiciel.....	7
7.1.1 Create (<i>création</i>)	7
7.1.2 Delete (<i>suppression</i>)	7
7.1.3 Deliver (<i>livraison</i>).....	7
7.1.4 Execute Program (<i>exécution de programme</i>).....	7
7.1.5 Get (<i>recherche</i>)	7
7.1.6 Install (<i>installation</i>).....	7
7.1.7 Revert (<i>désinstallation</i>).....	8
7.1.8 Set administrative state (<i>positionnement de l'état administratif</i>)	8
7.1.9 Terminate validation (<i>fin de validation</i>)	8
7.1.10 Validation (<i>validation</i>).....	8
7.1.11 Notifications.....	8
7.2 Autres fonctions	8
7.2.1 Backup (<i>sauvegarde</i>)	9
7.2.2 Restore (<i>restauration</i>)	9
7.3 Objet géré «Software Unit» (<i>unité de logiciel</i>)	9
7.3.1 Cycle de vie.....	9
7.3.2 Etats de l'objet géré «Software Unit».....	10
7.3.3 Opérations sur l'objet géré «Software Unit»	11
7.4 Objet géré «Executable Software» (<i>logiciel exécutable</i>)	11
7.4.1 Etats supplémentaires de l'objet géré «Executable Software»	11
7.4.2 Opérations supplémentaires pour l'objet géré «Executable Software»	12
7.5 Objet géré «Software Distributor» (<i>distributeur de logiciel</i>).....	12
7.5.1 Opérations sur l'objet géré «Software Distributor».....	12
7.5.2 Etats de l'objet géré «Software Distributor»	13
7.6 Relations	13
8 Définitions génériques	13
8.1 Objets gérés.....	13
8.1.1 Software (<i>logiciel</i>)	13
8.1.2 Software Unit (<i>unité de logiciel</i>).....	14
8.1.3 Executable Software (<i>logiciel exécutable</i>).....	15
8.1.4 Software Distributor (<i>distributeur de logiciel</i>)	15

	<i>Page</i>
8.2 Attributs	15
8.2.1 appliedPatches (<i>corrections appliquées</i>).....	15
8.2.2 checkSum (<i>somme de contrôle</i>)	16
8.2.3 dateDelivered (<i>date de livraison</i>).....	16
8.2.4 dateInstalled (<i>date d'installation</i>).....	16
8.2.5 dateOfCreation (<i>date de création</i>)	16
8.2.6 dateOfLastModification (<i>date de dernière modification</i>)	16
8.2.7 fileLocation (<i>emplacement de fichier</i>)	16
8.2.8 fileSize (<i>taille de fichier</i>).....	16
8.2.9 fileType (<i>type de fichier</i>).....	16
8.2.10 futureAutoBackupDestination (<i>destination de sauvegarde automatique future</i>)	16
8.2.11 futureAutoBackupTriggerThreshold (<i>seuil de déclenchement de sauvegarde automatique future</i>).....	17
8.2.12 futureAutoRestoreAllowed (<i>autorisation de restauration automatique future</i>)	17
8.2.13 futureAutoRestoreSource (<i>source de restauration automatique future</i>).....	17
8.2.14 identityOfCreator (<i>identité du créateur</i>)	17
8.2.15 identityOfLastModifier (<i>identité du dernier modificateur</i>)	17
8.2.16 lastBackupDestination (<i>dernière destination de sauvegarde</i>)	17
8.2.17 lastBackupTime (<i>date de la dernière sauvegarde</i>)	17
8.2.18 lastRestoreSource (<i>source de la dernière restauration</i>)	17
8.2.19 lastRestoreTime (<i>date de la dernière restauration</i>).....	17
8.2.20 noteField (<i>champ de note</i>)	17
8.2.21 softwareDistributorId (<i>identité du distributeur du logiciel</i>).....	17
8.3 Actions	18
8.3.1 backup (<i>sauvegarde</i>).....	18
8.3.2 deliver (<i>livraison</i>).....	18
8.3.3 execute program (<i>exécution de programme</i>).....	18
8.3.4 install (<i>installation</i>).....	18
8.3.5 restore (<i>restauration</i>)	18
8.3.6 revert (<i>désinstallation</i>)	19
8.3.7 terminateValidation (<i>fin de validation</i>)	19
8.3.8 validate (<i>validation</i>)	19
8.4 Notifications.....	19
8.4.1 autoBackupReport (<i>compte rendu de sauvegarde automatique</i>)	20
8.4.2 autoRestoreReport (<i>compte rendu de restauration automatique</i>)	20
8.4.3 deliverResultNotification (<i>notification de résultat de livraison</i>)	20
8.5 Paramètres.....	20
8.5.1 alarmEffectOnServiceParameter (<i>effet de l'alarme sur le service</i>)	20
8.5.2 softwareProcessingFailureParameter (<i>faute de traitement de logiciel</i>).....	20
8.6 Corrélations de nom	21
8.6.1 Sous-système – Software Distributor (<i>distributeur de logiciel</i>).....	21
8.6.2 Système – Software Distributor	21
8.6.3 Sous-système – Software Unit (<i>unité de logiciel</i>).....	21
8.6.4 Système – Software Unit.....	21
9 Définition de services	21
9.1 Introduction.....	21
9.2 Service Backup (<i>sauvegarde</i>)	21
9.3 Service Deliver (<i>livraison</i>).....	21
9.4 Service Execute program (<i>exécution de programme</i>).....	21
9.5 Service Install (<i>installation</i>).....	23
9.6 Service Restore (<i>restauration</i>)	24
9.7 Service Revert (<i>désinstallation</i>).....	24
9.8 Service Terminate validation (<i>fin de validation</i>).....	24
9.9 Service Validate (<i>validation</i>)	26
9.10 Service Auto backup report (<i>compte rendu de sauvegarde automatique</i>).....	26
9.11 Service Auto restore report (<i>compte rendu de restauration automatique</i>)	26
9.12 Service Deliver result notification (<i>notification de résultat de livraison</i>).....	27

	<i>Page</i>
10 Unités fonctionnelles.....	28
11 Protocole	28
11.1 Eléments de procédure	28
11.1.1 Procédure de sauvegarde.....	28
11.1.2 Procédure de livraison.....	29
11.1.3 Procédure d'exécution de programme	29
11.1.4 Procédure d'installation.....	30
11.1.5 Procédure de restauration.....	31
11.1.6 Procédure de désinstallation.....	31
11.1.7 Procédure de fin de validation	32
11.1.8 Procédure de validation.....	32
11.1.9 Procédure de compte rendu de sauvegarde automatique	33
11.1.10 Procédure de compte rendu de restauration automatique.....	34
11.1.11 Procédure de livraison de notification de résultat	34
11.2 Syntaxe abstraite	35
11.2.1 Objets	35
11.2.2 Paquetages.....	35
11.2.3 Attributs	36
11.2.4 Notifications.....	36
11.2.5 Actions	36
11.2.6 Corrélations de noms	37
11.3 Négociation des unités fonctionnelles.....	37
12 Relations avec d'autres fonctions	37
13 Conformité	38
13.1 Conformité statique.....	38
13.2 Conformité dynamique	38
13.3 Prescriptions concernant les déclarations de conformité d'implémentation de gestion.....	38
Annexe A – Définition de l'information de gestion.....	39
A.1 Managed Object Classes	39
A.1.1 Software	39
A.1.2 softwareUnit.....	39
A.1.3 executableSoftware	40
A.1.4 softwareDistributor	40
A.2 Packages.....	40
A.2.1 appliedPatchPackage.....	40
A.2.2 checkSumPackage.....	41
A.2.3 executeProgramPackage	41
A.2.4 fileInformationPackage.....	41
A.2.5 filePackage.....	41
A.2.6 informationAutoBackupPackage	41
A.2.7 informationAutoRestorePackage	41
A.2.8 informationBackupPackage	42
A.2.9 informationRestorePackage	42
A.2.10 installPackage	42
A.2.11 noteFieldPackage	42
A.2.12 processingErrorAlarmOnServicePackage	42
A.2.13 revertPackage	42
A.2.14 terminateValidationPackage	42
A.2.15 usageStatePackage	43
A.2.16 validationPackage	43

	<i>Page</i>
A.3 Attributes.....	43
A.3.1 appliedPatches.....	43
A.3.2 checkSum.....	43
A.3.3 dateDelivered	43
A.3.4 dateInstalled	44
A.3.5 dateOfCreation	44
A.3.6 dateOfLastModification	44
A.3.7 fileLocation	44
A.3.8 fileSize	45
A.3.9 fileType	45
A.3.10 futureAutoBackupDestination	45
A.3.11 futureAutoBackupTriggerThreshold	45
A.3.12 futureAutoRestoreAllowed	46
A.3.13 futureAutoRestoreSource	46
A.3.14 identityOfCreator	46
A.3.15 identityOfLastModifier	46
A.3.16 lastBackupDestination	47
A.3.17 lastBackupTime	47
A.3.18 lastRestoreSource	47
A.3.19 lastRestoreTime	47
A.3.20 noteField	48
A.3.21 softwareDistributorId	48
A.4 Name Bindings.....	48
A.4.1 softwareDistributor-subsystem	48
A.4.2 softwareDistributor-system	48
A.4.3 softwareUnit-subsystem	49
A.4.4 softwareUnit-system	49
A.5 Actions	49
A.5.1 deliver	49
A.5.2 executeProgram.....	49
A.5.3 install.....	50
A.5.4 revert	50
A.5.5 terminateValidation.....	51
A.5.6 validate.....	51
A.6 Notifications.....	51
A.6.1 autoBackupReport.....	51
A.6.2 autoRestoreReport.....	52
A.6.3 deliverResultNotification	52
A.7 Parameters.....	52
A.7.1 alarmEffectOnServiceParameter.....	52
A.7.2 softwareProcessingFailureParameter	53
A.8 Supporting Productions.....	53
A.9 Backup and Restore Actions	56
A.9.1 backup	56
A.9.2 restore.....	56
A.9.3 Backup Restore Supporting Productions	57
Annexe B – Formulaire MCS	58
B.1 Introduction.....	58
B.1.1 Purpose and structure	58
B.1.2 Instructions for completing the MCS proforma to produce an MCS	58
B.1.3 Symbols, abbreviations and terms	58
B.1.4 Table format.....	58
B.2 Identification of the implementation.....	59
B.2.1 Date of statement	59
B.2.2 Identification of the implementation	60
B.2.3 Contact	60

	<i>Page</i>
B.3 Identification of the Recommendations International Standards in which the management information is defined	60
B.3.1 Technical corrigenda implemented	60
B.3.2 Amendments implemented.....	60
B.4 Management conformance summary	60
Annexe C – Formulaire MICS.....	64
C.1 Introduction.....	64
C.2 Instructions for completing the MICS proforma to produce an MICS	64
C.3 Symbols, abbreviations and terms.....	64
C.4 Statement of conformance to the management information.....	64
C.4.1 Attributes.....	64
C.4.2 Actions	66
C.4.3 Notifications.....	71
C.4.4 Create and delete management operations.....	81
Annexe D – Formulaire MOCS.....	82
D.1 Introduction.....	82
D.2 Instructions for completing the MOCS proforma to produce a MOCS	82
D.3 Symbols, abbreviations and terms.....	82
D.4 Software unit managed object class	82
D.4.1 Statement of conformance to the managed object class.....	82
D.4.2 Packages.....	83
D.4.3 Attributes.....	84
D.4.4 Attribute groups	86
D.4.5 Actions	86
D.4.6 Notifications.....	90
D.4.7 Parameters.....	98
D.5 Executable software managed object class	98
D.5.1 Statement of conformance to the managed object class.....	98
D.5.2 Packages.....	99
D.5.3 Attributes.....	100
D.5.4 Attribute groups	102
D.5.5 Actions	103
D.5.6 Notifications.....	108
D.5.7 Parameters.....	117
D.6 Software distributor managed object class.....	117
D.6.1 Statement of conformance to the managed object class.....	117
D.6.2 Packages.....	118
D.6.3 Attributes.....	118
D.6.4 Attribute groups	118
D.6.5 Actions	119
D.6.6 Notifications.....	120
D.6.7 Parameters.....	122
Annex E – MRCS proforma for name binding.....	123
E.1 Introduction.....	123
E.2 Instructions for completing the MRCS proforma for name binding to produce an MRCS	123
E.3 Symbols, abbreviations and terms.....	123
E.4 Statement of conformance to the name binding.....	123
Annex F – MIDS proforma	125
F.1 Actions	125

	<i>Page</i>
Annex G – Recommendation M.3100 Software Object Class	126
G.1 Software Object Class.....	126
G.2 Packages.....	126
G.2.1 Administrative Operational States Package	126
G.2.2 Affected Object List Package	127
G.2.3 Attribute Value Change Notification Package.....	127
G.2.4 Create Delete Notifications Package.....	127
G.2.5 Current Problem List Package	127
G.2.6 Software Processing Error Alarm Package	127
G.2.7 State Change Notification Package.....	127
G.2.8 User Label Package.....	127
G.2.9 Vendor Name Package.....	127
G.2.10 Version Package.....	128
G.3 Attributes.....	128
G.3.1 Affected Object List.....	128
G.3.2 Alarm Status.....	128
G.3.3 Current Problem List.....	128
G.3.4 Software Id.....	129
G.3.5 User Label.....	130
G.3.6 Vendor Name	130
G.3.7 Version.....	130
Annex H – POSIX and Software Management Model Mapping	131

Résumé

Les systèmes d'exploitation (OS, *operations systems*), les éléments de réseaux (NE, *network elements*) et d'autres systèmes seront parfois soumis à des modifications de logiciels. Ces modifications peuvent porter sur le code de programme, comme c'est le cas par exemple pour une nouvelle version d'un programme lié au traitement d'appel ou pour la révision des tables de routage. Dans un réseau, il faut pouvoir gérer les logiciels qui peuvent être modifiés par certains systèmes d'exploitation distants. Dans ce contexte, la gestion peut comprendre: le contrôle du processus de téléchargement pour le transfert d'un produit logiciel d'un système d'exploitation à un autre système (système d'exploitation ou élément de réseau par exemple); le contrôle de l'installation du logiciel téléchargé dans le système; la capacité à vérifier la version du logiciel installé dans un système. La présente Recommandation | Norme internationale fournit un moyen, basé sur la messagerie X.700 normalisée, de gérer ce processus de modification de logiciels pour répondre aux besoins qui s'imposent. La présente Recommandation | Norme internationale ne spécifie pas de méthode de transfert de logiciel entre un système d'exploitation et un autre système et elle n'impose aucune contrainte concernant ces méthodes de transfert; dans la pratique, il pourrait s'agir d'un transfert de fichier, de l'envoi de messages électroniques ou de l'envoi de disquettes par courrier postal. Le rôle de la présente Recommandation | Norme internationale dans ce transfert est de gérer le lancement de la livraison de logiciels à un autre système en conservant les informations sur le lancement d'une telle livraison et en reliant ces informations à la réception effective, indiquée par un avis de livraison effectuée envoyé par le système de réception.

NORME INTERNATIONALE**RECOMMANDATION UIT-T****TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – GESTION-SYSTÈMES: FONCTION DE GESTION DE LOGICIEL****1 Domaine d'application**

La fonction de gestion de logiciel comprend la gestion d'un système en ce qui concerne la livraison du logiciel ainsi que la gestion du logiciel au sein d'un système.

Il faut distinguer deux aspects du logiciel, décrits comme l'aspect «dormant» et l'aspect «actif» du logiciel.

L'aspect dormant est lié aux données stockées dans un système géré et à la manière dont elles sont livrées et installées. Les données sont en général des informations stockées telles que des fichiers de données ou des tables, mais il peut s'agir également de fichiers contenant du code exécutable. Le domaine d'application de la présente Recommandation | Norme internationale concerne l'aspect dormant du logiciel.

L'aspect actif du logiciel est lié aux ressources qui utilisent le logiciel. Il n'existe pas de différence réelle entre cet aspect et la vue normale de la gestion des ressources. La présente Recommandation | Norme internationale ne concerne pas l'aspect actif du logiciel. Toutefois, les relations entre les objets gérés qui représentent des ressources utilisant le logiciel et les objets gérés représentant le logiciel utilisé, c'est-à-dire l'aspect dormant du logiciel, font partie du domaine d'application de la présente Recommandation | Norme internationale.

Le domaine d'application de la présente Recommandation | Norme internationale englobe les points suivants:

- initialisation du transfert du logiciel;
- gestion du logiciel après le transfert;
- activation du logiciel, y compris la gestion des versions et des corrections;
- désactivation du logiciel;
- inversion de changements apportés au logiciel;
- validation du logiciel;
- interrogation du logiciel;
- sauvegarde du logiciel;
- restauration du logiciel.

NOTE 1 – La sauvegarde et la restauration sont définies d'une manière générique de sorte qu'elles peuvent s'appliquer à des ressources autres que du logiciel.

Le domaine d'application de la présente Recommandation | Norme internationale ne comprend pas les points suivants:

- mécanisme de transfert du logiciel;
- stockage physique du logiciel (mise en correspondance entre le logiciel et le stockage physique du fichier sur disquette, disque dur, etc.);
- formatage du logiciel;
- nomenclature des produits logiciels;
- l'ordonnancement de commandes de gestion de logiciel;
- supervision du logiciel;
- gestion des processus exécutés sur un système.

NOTE 2 – La prise en charge d'activités de redevance de licence (c'est-à-dire la facturation et la comptabilisation) et de trace du logiciel (c'est-à-dire la trace de l'exécution du logiciel à des fins de mise au point) est en dehors du domaine d'application de la présente Recommandation | Norme internationale et appelle une étude ultérieure.

2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valable pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes internationales indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations de l'UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.210 (1993) | ISO/CEI 10731:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: conventions pour la définition des services de l'interconnexion des systèmes ouverts.*
- Recommandation X.701 du CCITT (1992) | ISO/CEI 10040:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Aperçu général de la gestion des systèmes*¹⁾.
- Recommandation X.720 du CCITT (1992) | ISO/CEI 10165-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: modèle d'information de gestion.*
- Recommandation X.721 du CCITT (1992) | ISO/CEI 10165-2:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: définition des informations de gestion.*
- Recommandation X.722 du CCITT (1992) | ISO/CEI 10165-4:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: directives pour la définition des objets gérés.*
- Recommandation UIT-T X.723 (1993) | ISO/CEI 10165-5:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: informations génériques de gestion.*
- Recommandation UIT-T X.724 (1993) | ISO/CEI 10165-6:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'instances associées à la gestion OSI.*
- Recommandation X.730 du CCITT (1992) | ISO/CEI 10164-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des objets.*
- Recommandation X.731 du CCITT (1992) | ISO/CEI 10164-2:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion d'état.*
- Recommandation X.732 du CCITT (1992) | ISO/CEI 10164-3:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: attributs relationnels.*
- Recommandation UIT-T X.738 (1993) | ISO/CEI 10164-13:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de récapitulation.*
- Recommandation UIT-T X.739 (1993) | ISO/CEI 10164-11:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: objets et attributs métriques.*
- Recommandation X.740 du CCITT (1992) | ISO/CEI 10164-8:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de piste de vérification de sécurité.*
- Recommandation UIT-T X.741 (1995) | ISO/CEI 10164-9:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: objets et attributs de contrôle d'accès.*
- Recommandation UIT-T X.742 (1995) | ISO/CEI 10164-10:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de comptage d'utilisation aux fins de comptabilité.*

¹⁾ Amendée par la Rec. UIT-T X.701/Cor.2 (1995) | ISO/CEI 10040/Cor.2:1995.

- Recommandation UIT-T X.745 (1993) | ISO/CEI 10164-12:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des tests.*
- Recommandation UIT-T X.746 (1995) | ISO/CEI 10164-15:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de programmation.*

2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique

- Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*
ISO/CEI 8824:1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de la notation de syntaxe abstraite numéro 1 (ASN.1).*
- Recommandation X.209 du CCITT (1988), *Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1).*
ISO/CEI 8825:1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de règles de base pour coder la notation de syntaxe abstraite UNE (ASN.1).*
- Recommandation UIT-T X.291 (1995), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Spécification de suite de tests abstraite.*
ISO/CEI 9646-2:1994, *Technologies de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 2: Spécification des suites de tests abstraites.*
- Recommandation UIT-T X.296 (1995), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Déclarations de conformité d'instance.*
ISO/CEI 9646-7:1995, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Essais de conformité – Méthodologie générale et procédures – Partie 7: Déclaration de conformité des mises en œuvre.*
- Recommandation X.700 du CCITT (1992), *Cadre de gestion pour l'interconnexion de systèmes ouverts pour les applications du CCITT.*
ISO/CEI 7498-4:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 4: Cadre général de gestion.*
- Recommandation X.710 du CCITT (1991), *Définition du service commun de transfert d'informations de gestion pour les applications du CCITT.*
ISO/CEI 9595:1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Définition du service commun d'informations de gestion.*
- Recommandation X.711 du CCITT (1991), *Spécification du protocole commun de transfert d'informations de gestion pour les applications du CCITT.*
ISO/CEI 9596-1:1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Protocole commun d'information de gestion – Partie 1: Spécification.*

2.3 Références additionnelles

- Recommandation UIT-T M.3100 (1995), *Modèle générique d'information de réseau.*
- Recommandation UIT-T M.3101 (1995), *Déclarations de conformité d'objets gérés pour le modèle générique d'information de réseau.*
- ISO/CEI 15068-2²⁾, *Technologies de l'information – Administration de système POSIX (Portable Operating System Interface) – Partie 2: Gestion du logiciel (IEEE P1387.2).*

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

²⁾ Actuellement à l'état de projet.

3.1 Définitions du cadre général de gestion

La présente Recommandation | Norme internationale utilise le terme ci-après, défini dans la Rec. X.700 du CCITT et l'ISO/CEI 7498-4:

- objet géré.

3.2 Définitions de l'aperçu général de la gestion-systèmes

La présente Recommandation | Norme internationale utilise les termes ci-après, définis dans la Rec. X.701 du CCITT | ISO/CEI 10040:

- a) classe d'objets gérés;
- b) déclaration de conformité d'information de gestion (MICS);
- c) déclaration de conformité d'objet géré (MOCS);
- d) formulaire de déclaration MICS;
- e) formulaire de déclaration MOCS;
- f) notification.

3.3 Définitions du service CMIS

La présente Recommandation | Norme internationale utilise le terme ci-après, défini dans la Rec. X.710 du CCITT et l'ISO/CEI 9595:

- attribut.

3.4 Définitions du modèle d'information de gestion

La présente Recommandation | Norme internationale utilise les termes ci-après, définis dans la Rec. X.720 du CCITT | ISO/CEI 10165-1:

- a) action;
- b) comportement;
- c) corrélation de nom;
- d) paquetage (ou lot de propriétés);
- e) hyperclasse.

3.5 Directives pour l'établissement de la définition des objets gérés

La présente Recommandation | Norme internationale utilise le terme ci-après, défini dans la Rec. X.722 du CCITT | ISO/CEI 10165-4:

- squelette (ou modèle).

3.6 Définitions des formulaires de déclaration de conformité d'implémentation

La présente Recommandation | Norme internationale utilise les termes ci-après, définis dans la Rec. X.724 du CCITT | ISO/CEI 10165-6:

- a) déclaration de conformité de relation gérée (MRCS);
- b) récapitulatif de conformité de gestion (MCS);
- c) formulaire de récapitulatif MCS;
- d) formulaire de déclaration MRCS.

3.7 Définitions supplémentaires

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

3.7.1 sauvegarde (*backup*): l'opération de sauvegarde a pour effet, lorsqu'elle s'applique à des objets gérés logiciels, de procéder à une copie des ressources sous-jacentes. Elle n'a pas d'effet direct en ce qui concerne les ressources logicielles d'origine.

NOTE – L'opération de sauvegarde peut également s'appliquer à d'autres classes d'objets gérés, toutefois seule l'application au logiciel est prise en considération pour ce qui est de la présente Recommandation | Norme internationale.

3.7.2 livraison (*deliver*): la livraison du logiciel est le processus de mise en place simultanée sur le système d'un objet logiciel géré et des ressources qui lui sont associées.

NOTE – La manière selon laquelle ceci est réalisé, et en particulier le choix du mécanisme de transfert, est en dehors du domaine d'application de la présente Recommandation | Norme internationale.

3.7.3 logiciel exécutable (*executable*): le logiciel exécutable est un logiciel pouvant être exécuté d'une certaine manière. Il peut être possible de faire exécuter un logiciel exécutable au moyen d'une commande de gestion, toutefois il peut également être possible de n'exécuter le logiciel qu'au moyen d'une action locale.

3.7.4 installation (*install*): l'installation du logiciel met celui-ci sous une forme convenant à son utilisation, par opposition à la forme sous laquelle il peut être transféré entre systèmes au cours de l'opération de livraison. Une fois installé, le logiciel peut être utilisé par d'autres objets gérés.

NOTE – L'installation peut également impliquer la mise en place de relations entre objets logiciels gérés.

3.7.5 restauration (*restore*): l'opération de restauration peut être utilisée pour remplacer les ressources associées à un objet logiciel géré par une copie faite lors d'une sauvegarde antérieure.

NOTE – Comme pour la sauvegarde, l'opération de restauration peut également s'appliquer à d'autres classes d'objets gérés, toutefois seule l'application au logiciel est prise en considération pour ce qui est de la présente Recommandation | Norme internationale.

3.7.6 désinstallation (*revert*): l'opération de désinstallation défait le résultat d'une opération d'installation antérieure ou supprime une correction antérieure.

3.7.7 utilisation (*utilize*): l'utilisation est décrite par une relation entre des processus actifs et des ressources logicielles. Elle représente le fait que le processus utilise le logiciel. Le logiciel ne peut être utilisé que s'il possède un état opérationnel «ENABLED» (en service) et un état administratif «UNLOCKED» (déverrouillé). Il se peut qu'il existe des contraintes concernant le nombre et la nature des utilisateurs éventuels de la ressource logicielle. Lorsqu'un logiciel est utilisé, son état d'utilisation sera «ACTIVE» (actif) ou «BUSY» (occupé).

NOTE – Ces relations ne font pas partie du domaine d'application de la présente Recommandation | Norme internationale en ce qui concerne l'aspect de gestion.

3.7.8 validation (*validate*): l'opération de validation est utilisée pour vérifier l'intégrité des ressources logicielles.

NOTE – La façon de procéder à cette vérification est une question locale.

4 Symboles et abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes sont utilisées:

CMIP	Protocole commun d'information de gestion (<i>common management information protocol</i>)
Conf	Confirmation (<i>confirmation</i>)
ICS	Déclaration de conformité d'implémentation (<i>implementation conformance statement</i>)
Id	Identificateur
MCS	Récapitulatif de conformité de gestion (<i>management conformance summary</i>)
MICS	Déclaration de conformité d'information de gestion (<i>management information conformance statement</i>)
MOCS	Déclaration de conformité d'objet géré (<i>managed object conformance statement</i>)
MRCS	Déclaration de conformité de relation gérée (<i>managed relationship conformance statement</i>)

PICS	Déclaration de conformité d'implémentation de protocole (<i>protocol implementation conformance statement</i>)
POSIX	Interface portable de système d'exploitation (<i>portable operating system interface</i>)

5 Conventions

La présente Spécification définit des services pour établir et mettre fin à une relation de sauvegarde selon les conventions de description définies dans la Rec. UIT-T X.210 | ISO/CEI 10731, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base – Conventions pour la définition des services OSI*.

La notation suivante est utilisée dans la présente Recommandation | Norme internationale pour les tableaux de paramètres de service:

- M le paramètre est obligatoire;
- (=) la valeur du paramètre est égale à celle du paramètre de la colonne de gauche;
- U l'utilisation du paramètre est une option d'utilisateur de service;
 - le paramètre ne figure pas dans l'interaction décrite par la primitive;
- C le paramètre est conditionnel;
- P le paramètre est soumis aux contraintes imposées par la Rec. X.710 du CCITT et l'ISO/CEI 9595.

6 Prescriptions pour la gestion de logiciel

La gestion du logiciel doit être capable de satisfaire aux prescriptions suivantes, compte tenu de commandes ou de conditions pouvant être imposées:

- a) capacité de demander une livraison de logiciel à un système de gestion spécifié;
- b) capacité de gérer l'installation de logiciel sur un système géré, y compris l'installation de corrections (par exemple, des améliorations) et la possibilité de revenir à une version antérieure du logiciel;
- c) capacité de lancer l'exécution d'un programme logiciel;
- d) capacité d'interrogation de la valeur des attributs pour tous les logiciels détenus par un système géré;
- e) capacité de création et de suppression de logiciel détenu par un système géré;
- f) capacité de validation de logiciel détenu par un système géré afin d'en vérifier l'intégrité, ainsi que la capacité de mettre fin à la validation;
- g) capacité de restreindre, à des fins administratives, l'utilisation de ressources logicielles d'un système géré;
- h) capacité de sauvegarder un item logiciel et de restaurer un item logiciel sauvegardé précédemment.

Le modèle ne fera pas obstacle à l'utilisation de procédures d'archivage, de comptabilité, de vérification et de gestion de licence lors de chaque étape de ce processus.

La réussite ou l'échec de l'opération doit dans tous les cas faire l'objet d'un compte rendu au système gestionnaire.

La manière dont des ressources logicielles utilisent d'autres ressources logicielles est représentée par des relations entre objets logiciels gérés. Ce type de relation, appelé «utilisation», est employé pour indiquer quelles sont les versions de logiciel devant être utilisées à un instant donné. La gestion de cette relation appelle une étude ultérieure.

7 Modèle concernant la fonction de gestion de logiciel

La fonction de gestion de logiciel s'intéresse en premier lieu à la gestion de composants du logiciel. Dans ce contexte, le logiciel comprend des données, de l'information de commande et des instructions exécutables. La vue de gestion du logiciel peut être représentée par un objet géré de la classe «**Software Unit**» (*unité de logiciel*), la gestion d'un logiciel représentant des instructions exécutables possède des caractéristiques supplémentaires, de sorte qu'une autre classe «**Executable Software Unit**» (*unité de logiciel exécutable*) est définie comme sous-classe de la classe unité de logiciel.

Un autre aspect de la gestion de logiciel est la gestion de la livraison de logiciel au système géré. Le logiciel n'est pas nécessairement géré et livré dans les mêmes unités. Il est par exemple possible qu'un certain nombre d'unités de logiciel soient livrées ensemble à un système géré (par exemple au moyen d'un CD-ROM). De même, si une unité de logiciel de taille importante doit être livrée au moyen d'un réseau de communication, il peut être nécessaire de la découper en parties plus petites pour effectuer la livraison. Un autre cas est celui d'une fourniture limitée à la livraison de modifications exigées par un objet logiciel existant, par exemple une correction. La gestion de la livraison est faite au moyen de l'objet géré «Software Distributor» (*distributeur de logiciel*).

7.1 Fonctions de gestion de logiciel

7.1.1 Create (*création*)

Une unité de logiciel peut être créée sur un système géré au moyen de l'opération «Create». Une unité de logiciel peut être créée dans l'état «créé» ou dans l'état «livré» compte tenu du comportement. Une unité de logiciel peut être dupliquée ou copiée au moyen d'une opération de gestion «Create with Reference Object» (*création avec objet de référence*) qui fait référence à cet objet géré «unité de logiciel». Une unité de logiciel peut également être créée comme résultat d'une opération de livraison effectuée sur un objet géré «distributeur de logiciel» ou comme résultat d'une action locale. Un objet géré «unité de logiciel» peut émettre une notification de création d'objet à la suite de sa création.

7.1.2 Delete (*suppression*)

Une unité de logiciel peut être supprimée sur un système géré au moyen de l'opération «Delete». L'opération de suppression d'une unité de logiciel a pour effet de bord la suppression des ressources sous-jacentes associées.

7.1.3 Deliver (*livraison*)

Il est possible de demander la livraison d'un ensemble coordonné d'unités de logiciel. Le résultat de la livraison indique un succès ou un échec. La livraison est effectuée en émettant l'action «Deliver» à destination de l'objet géré «distributeur de logiciel». Le résultat d'une action de livraison est la fourniture d'une copie des items de logiciel cible, se trouvant dans l'état «livré», à destination du système cible. Un effet de bord de cette opération peut être la création d'un ou de plusieurs objets associés (par exemple des unités logicielles).

Le conditionnement des unités logicielles et le choix du mécanisme de transfert sont du ressort local et sortent du cadre de la présente Recommandation | Norme internationale. Cette information peut, par exemple, être préconfigurée ou spécifiée dans l'action de remise en même temps que les autres informations associées.

7.1.4 Execute Program (*exécution de programme*)

Un système gestionnaire peut demander l'exécution de logiciel exécutable au moyen de l'opération «execute program». La présente Recommandation | Norme internationale ne spécifie pas comment une telle exécution du logiciel peut être gérée par la suite, mais se borne à la fourniture d'un mécanisme de lancement de l'exécution du logiciel exécutable.

7.1.5 Get (*recherche*)

L'opération «Get» fournit la possibilité de trouver une information concernant le logiciel, par exemple de déterminer quel logiciel est présent, quel logiciel est disponible pour une utilisation, quelles sont les relations entre logiciels, etc. En cas de succès, le résultat de l'opération de recherche fournit l'information demandée.

7.1.6 Install (*installation*)

L'opération d'installation adapte le logiciel pour une utilisation particulière. Cette opération peut nécessiter potentiellement un volume et un temps de traitement importants. Elle peut entraîner la vérification de l'existence de tous les composants de la version du logiciel cible sur le système géré (comme partie constituante d'une mise à jour ou préexistants sur le système), ainsi que leur assemblage pour les préparer pour une utilisation. L'installation est réalisée par l'envoi de l'action «Install» à destination de l'objet géré «unité de logiciel» dont elle constitue une caractéristique optionnelle.

Une correction est une modification d'un logiciel pouvant être représentée par un objet géré logiciel. En conséquence, une correction peut être livrée, installée, utilisée et copiée au moyen de la gestion du logiciel.

L'installation d'une correction représente un cas spécial dans la mesure où la correction est livrée et où l'application de la correction en constitue l'installation et a pour résultat la production d'une version mise à jour du logiciel, prête à l'utilisation. La version mise à jour reçoit à des fins d'identification un numéro de version mis à jour qui est fourni en même temps que la correction. Le résultat d'une installation en indique le succès ou l'échec.

L'installation peut commencer après une livraison complète du logiciel. Il est possible que l'installation puisse nécessiter une coordination qui est toutefois en dehors du domaine d'application de la présente Recommandation | Norme internationale. Des exemples de coordination sont:

- l'obligation de terminer la livraison de plusieurs items de logiciel avant qu'un autre item de logiciel puisse être installé;
- la vérification que la version actuelle du logiciel n'est pas utilisée à l'instant donné avant l'activation d'une nouvelle version;
- la vérification que des versions particulières de versions d'autres logiciels sont installées ou non;
- la synchronisation de l'installation avec d'autres systèmes ouverts, la possibilité d'installations simultanées mais indépendantes sur plus d'un système ouvert, la possibilité de lier des installations sur plusieurs systèmes ouverts pour constituer une seule unité de travail, etc.

7.1.7 Revert (*désinstallation*)

L'opération «Revert» est utilisée pour faire revenir un objet géré «unité de logiciel» dans un état où il n'est pas installé ou pour effectuer la suppression de l'application d'une ou de plusieurs corrections. Les corrections à supprimer sont identifiées par un nom de fichier ou par un identificateur d'une instance de l'objet géré «unité de logiciel». L'opération de désinstallation est réalisée par l'envoi d'une action à destination de l'objet géré «unité de logiciel» dont elle constitue une caractéristique optionnelle.

Il peut être nécessaire de gérer une information supplémentaire indiquant de quelle manière un objet géré «unité de logiciel» peut être désinstallé. Ce point est toutefois en dehors du domaine d'application de la présente Recommandation | Norme internationale.

7.1.8 Set administrative state (*positionnement de l'état administratif*)

Une fois installé, c'est-à-dire personnalisé pour une utilisation particulière, le logiciel peut être rendu disponible pour une utilisation au moyen de l'envoi de l'opération «positionnement de l'état administratif sur déverrouillé» à destination des objets gérés «logiciel exécutable». Il est possible de modifier l'état de disponibilité du logiciel en positionnant l'état administratif de l'objet géré «unité de logiciel» sur «fermeture en cours» ou sur «verrouillé».

7.1.9 Terminate validation (*fin de validation*)

La validation du logiciel peut être interrompue au moyen de l'opération «Terminate validation». L'opération de fin de validation est réalisée par l'envoi d'une action à destination de l'objet géré «unité de logiciel» dont elle constitue une caractéristique optionnelle. Une validation peut être terminée d'une manière prématuée car elle peut prendre potentiellement un temps important.

7.1.10 Validation (*validation*)

L'intégrité d'un logiciel peut être vérifiée au moyen de l'opération «Validation». Il est également possible de valider le fait qu'un logiciel livré précédemment reste dans un état permettant son utilisation. Le résultat de la validation indique si le logiciel a été validé ou non. L'opération de validation est réalisée par l'envoi d'une action à destination de l'objet géré «unité de logiciel» dont elle constitue une caractéristique optionnelle.

7.1.11 Notifications

Toutes les opérations pouvant s'appliquer à une unité de logiciel peuvent exiger que des résultats soient envoyés au système gestionnaire en même temps que la confirmation de l'exécution de l'opération. Il peut également être nécessaire de notifier à un autre système gestionnaire la fin de l'exécution d'une de ces opérations lorsque la demande concernant cette opération n'a pas été émise par le système gestionnaire.

7.2 Autres fonctions

Les fonctions de sauvegarde et de restauration ont un caractère générique et sont conçues pour être utilisées pour des objets gérés autres que les objets logiciels concernés par la présente Recommandation | Norme internationale.

7.2.1 Backup (*sauvegarde*)

L'opération «Backup» peut être utilisée par un système gestionnaire pour demander l'exécution de la sauvegarde d'un objet cible. Lorsqu'elle s'applique à un objet géré logiciel, l'opération de sauvegarde a pour effet de faire réaliser une copie des ressources sous-jacentes. Elle n'a pas d'effet direct sur les ressources logicielles originales.

L'opération de sauvegarde peut également s'appliquer à d'autres classes d'objets gérés, en ce qui concerne toutefois la présente Recommandation | Norme internationale, seule l'utilisation concernant le logiciel est prise en considération.

7.2.2 Restore (*restauration*)

L'opération «Restore» peut être utilisée par un système gestionnaire pour demander l'exécution de la restauration d'un objet cible précédemment sauvegardé.

L'opération de restauration peut également s'appliquer à d'autres classes d'objets gérés, en ce qui concerne toutefois la présente Recommandation | Norme internationale, seule l'utilisation concernant le logiciel est prise en considération.

7.3 Objet géré «Software Unit» (*unité de logiciel*)

7.3.1 Cycle de vie

Lorsqu'un objet géré «Software Unit» (*unité de logiciel*) est créé, il représente une ressource qui est disponible pour accepter la livraison d'un composant de logiciel.

Il existe quatre mécanismes ou opérations permettant de créer un objet géré «unité de logiciel»:

- 1) le logiciel peut être reproduit sur un système géré cible en l'absence de toute opération, par exemple par un moyen local;
- 2) l'utilisation de l'opération «Create» (*création*) permet de créer une unité de logiciel dans l'état «livré»;
- 3) l'utilisation de l'opération «Create with Reference Object» (*création avec un objet de référence*) faisant référence à un objet géré «unité de logiciel» source et qui est émise vers une destination cible permet la copie ou la reproduction du logiciel source;
- 4) l'utilisation de l'opération «Delivery» (*livraison*) émise à destination de l'objet géré «Software Distributor» (*distributeur de logiciel*) du système source fournissant la possibilité de créer des objets gérés «logiciel exécutable» au sein de la destination cible dans l'état «livré». L'opération de livraison appliquée à un objet géré «distributeur de logiciel» a pour effet la livraison du logiciel à la destination cible. Lorsque l'ensemble du logiciel exigé a été livré avec succès, une notification de résultat de livraison est émise par l'objet géré «distributeur de logiciel» pour indiquer que la livraison a été réalisée.

Les opérations de création et de création avec objet de référence ont pour effet la création d'un objet géré «unité de logiciel» sur la cible destinataire. La réussite de la création sera indiquée par l'existence d'un objet géré «unité de logiciel» sur la destination cible au moyen d'une réponse de création ou d'une notification de création d'objet.

S'il a été créé précédemment mais non livré, l'objet géré «unité de logiciel» peut émettre une notification de changement d'état lorsque la livraison est terminée, ce qui indique que son état est passé de l'état «créé» à «livré». Un objet géré «unité de logiciel» peut également être instancié dans l'état «livré».

Une fois que le logiciel se trouve dans l'état «livré», la prochaine étape pour ce logiciel est de passer dans l'état installé. L'installation comporte la personnalisation du logiciel à des fins d'utilisation. L'installation peut par exemple impliquer la préparation et l'adaptation de l'unité de logiciel pour une amélioration, éventuellement en réalisant une copie des ressources logicielles en vigueur et en appliquant les modifications à la copie. L'installation peut impliquer la mise en place de relations de configuration et de dépendance entre l'unité de logiciel et d'autres objets logiciels.

Une fois le logiciel installé, il peut être utilisé par d'autres objets gérés. Il peut être nécessaire de rendre le logiciel disponible en positionnant l'état administratif sur «déverrouillé» si ce dernier se trouvait dans l'état «verrouillé». Cette opération est implémentée en appliquant l'opération de remplacement à l'attribut «Administrative State» (*état administratif*) défini dans la Rec. X.731 du CCITT | ISO/CEI 10164-2.

Le logiciel est rendu indisponible pour une utilisation par verrouillage de l'état administratif, c'est-à-dire en le positionnant sur «verrouillé» ou «fermeture en cours». Ceci interdit à tout nouveau processus l'utilisation du logiciel. Si l'état administratif est positionné sur verrouillé, tous les processus actifs à l'instant donné sont abandonnés. Si l'état administratif est positionné sur fermeture en cours, les processus actifs peuvent continuer à s'exécuter, mais le lancement

de nouveaux processus n'est plus autorisé. L'état administratif est automatiquement positionné sur verrouillé lorsque tous les processus actifs ont terminé leur exécution. Cette opération est implémentée en appliquant l'opération de remplacement à l'attribut «Administrative State» défini dans la Rec. X.731 du CCITT | ISO/CEI 10164-2.

L'opération «Revert» (*désinstallation*) peut être utilisée pour inverser l'effet d'une opération d'installation antérieure ou pour supprimer l'application d'une correction.

L'opération «Validate» (*validation*) effectue une vérification de l'intégrité du logiciel, par exemple en exécutant un algorithme de contrôle par sommation ou une détection de virus. Il peut également être possible de mettre fin prématurément au processus de validation en utilisant l'opération de fin de validation.

La sauvegarde du logiciel au moyen de l'opération «Backup» est considérée comme la recopie d'une copie du logiciel en exploitation en vue d'une utilisation en cas de faute de la version en vigueur. Le logiciel sauvegardé peut être restauré au moyen de l'opération «Restore» (*restauration*).

7.3.2 Etats de l'objet géré «Software Unit»

Un objet géré «unité de logiciel» peut se trouver dans l'un des états du cycle de vie en fonction de la dernière opération qui lui a été appliquée. Les états possibles et leurs significations sont:

- créé (*created*): la livraison du logiciel n'est pas terminée, toutefois certaines ressources arbitraires du système géré ont été allouées à l'unité de logiciel;
- livré (*delivered*): le logiciel a été livré avec succès au système géré;
- installé (*installed*): le logiciel a été installé avec succès sur le système géré.

Le Tableau 1 donne la correspondance entre les états d'un objet géré «unité de logiciel» et les valeurs d'état et de statut définies dans la Rec. X.731 du CCITT | ISO/CEI 10164-2.

Tableau 1 – Etats du cycle de vie de l'unité de logiciel

Etat de l'unité de logiciel	Valeur du statut procédural {Initialization Required}	Valeur du statut de disponibilité {Not Installed}
Créé (<i>Created</i>)	Présent	Présent
Livré (<i>Delivered</i>)	Absent	Présent
Installé (<i>Installed</i>)	Absent	Absent

Pour l'objet géré «unité de logiciel», les attributs d'état «Administrative State», «Operational State», «Procedural Status» et «Availability Status» (*respectivement état administratif, état opérationnel, statut procédural et statut de disponibilité*) sont obligatoires, alors que l'attribut «Usage State» (*état d'utilisation*) est facultatif.

Les états «créé», «livré» et «installé» s'excluent mutuellement, c'est-à-dire qu'une unité de logiciel doit être à un instant donné dans un seul de ces états à la fois. Les états de validation et de faute sont indépendants des états «créé», «livré» et «installé».

Les états «validation en cours» et «en faute» sont des états secondaires qui peuvent également exister dans le cycle de vie. L'état «validation en cours» correspond à un statut de disponibilité qui contient la valeur {In Test} (en cours de test). Un objet géré «unité de logiciel» peut entrer dans l'état «validation en cours» ou le quitter indépendamment de la valeur de ses autres états. Une unité de logiciel peut par exemple se trouver dans l'état «validation en cours» pendant une faute (voir le Tableau 2).

Tableau 2 – Etat de validation de l'unité de logiciel

Etat de l'unité de logiciel	Valeur du statut de disponibilité {In Test}
Validation en cours (<i>Validating</i>)	Présent

Une unité de logiciel qui se trouve dans l'état «créé», «installé» ou «validé» peut également se trouver dans l'état «en faute». L'état «en faute» correspond à un statut de disponibilité qui contient la valeur {Failed}. Un objet géré «unité de logiciel» peut entrer dans l'état «en faute» ou le quitter indépendamment de la valeur de ses autres états. Une opération de validation peut par exemple avoir pour effet de faire passer l'unité de logiciel dans l'état «en faute» tout en restant dans l'état «validation en cours», ou une unité logicielle peut quitter l'état «en faute» à la suite d'une opération de désinstallation (voir le Tableau 3).

Tableau 3 – Etat de faute de l'unité de logiciel

Etat de l'unité de logiciel	Valeur du statut de disponibilité {Failed}
En faute (<i>Failed</i>)	Présent

7.3.3 Opérations sur l'objet géré «Software Unit»

Un certain nombre d'opérations associées à un objet géré «unité de logiciel» sont utilisées pour en modifier l'état.

Ces opérations sont:

- Backup (*sauvegarde*): sauvegarde le logiciel vers la destination cible.
- Create (*création*): crée un nouvel objet géré «unité de logiciel» sur le système géré.
- Delete (*suppression*): supprime l'objet géré «unité de logiciel» sur le système géré et peut également avoir pour effet la suppression des ressources associées.
- Install (*installation*): prépare le logiciel pour une utilisation.
- Restore (*restauration*): restaure un logiciel sauvegardé à partir de la destination cible.
- Revert (*désinstallation*): inverse l'application d'une correction ou une installation.
- Terminate Validation (*fin de validation*): met fin au processus de validation du logiciel.
- Validate (*validation*): vérifie l'intégrité du logiciel.

Les opérations «Create» et «Delete» correspondent aux opérations normalisées Create et Delete définies dans la Rec. X.720 du CCITT | ISO/CEI 10165-1, les autres correspondent à des actions. En outre les valeurs des attributs peuvent être demandées et modifiées en utilisant les opérations «Get» et «Replace» définies dans la Rec. X.720 du CCITT | ISO/CEI 10165-1.

Le Tableau 4 donne les transitions autorisées entre les états. Le Tableau 4 ne s'intéresse qu'aux opérations pouvant affecter les états, les opérations qui n'entraînent pas de changement d'état ne sont pas représentées.

7.4 Objet géré «Executable Software» (*logiciel exécutable*)

La classe d'objets gérés «Executable Software» est une sous-classe de la classe d'objets gérés «Software Unit» (*unité de logiciel*) possédant des caractéristiques supplémentaires qui décrivent les fonctionnalités liées à la possibilité d'exécution. Un logiciel exécutable est un logiciel pouvant être exécuté, par exemple à l'aide de moyens locaux ou au moyen de l'opération d'exécution. Il peut être possible d'effectuer l'exécution de logiciel exécutable par une commande de gestion, il est toutefois possible que l'exécution ne puisse être effectuée que par des moyens locaux.

Bien que ce point soit en dehors du domaine d'application de la présente Recommandation | Norme internationale, le modèle n'exclut pas la possibilité de sous-classes définissant une information telle que la possibilité d'utilisateurs uniques ou multiples, les conditions d'activité ou d'occupation du logiciel, ou la possibilité d'autoriser un nombre maximal d'utilisateurs multiples.

7.4.1 Etats supplémentaires de l'objet géré «Executable Software»

L'état d'utilisation défini dans la Rec. X.731 du CCITT | ISO/CEI 10164-2 indique si le logiciel exécutable est en cours d'utilisation à un instant donné.

Tableau 4 – Matrice de transitions d'état de l'unité de logiciel

Etats	Inexistant	Créé	Livré	Installé
Inexistant	N/A	Créé ^{a)} Créé avec référence ^{a)} ou Moyen local	Créé ^{a)} Créé avec référence ^{a)} ou Livré par le logiciel	N/A
Créé	Supprimé	–	Livré par le logiciel	N/A
Livré	Supprimé	N/A	–	Installé ou moyen local ^{a)}
Installé	Supprimé	N/A	Désinstallation ^{a)}	Désinstallation ou – ^{a)}

^{a)} Fonction du comportement de l'objet ou de moyens locaux.
N/A Changement d'état impossible.
– Toute opération ne causant pas de changement d'état peut s'appliquer, par exemple la sauvegarde, la validation, la fin de validation, l'interrogation ou la restauration.

L'état d'utilisation est un attribut obligatoire de la classe d'objets gérés «logiciel exécutable» en plus de ceux qui sont obligatoires pour la classe d'objets gérés «logiciel exécutable». Les valeurs de «repos», «actif» et «occupé» sont autorisées pour l'état d'utilisation de la classe d'objets gérés «logiciel exécutable».

7.4.2 Opérations supplémentaires pour l'objet géré «Executable Software»

Les opérations suivantes s'appliquent à un objet géré «logiciel exécutable» en plus de celles définies pour l'objet géré «unité de logiciel»:

- Execute Program (*exécution de programme*): cause l'exécution de l'objet géré logiciel.

L'opération d'exécution de programme est utilisée pour lancer l'exécution du programme logiciel représenté par le logiciel exécutable. Celui-ci doit avoir été installé avant d'être exécuté.

L'état d'un logiciel après son exécution sera déterminé par le comportement de l'instance spécifique du logiciel exécutable ou par un moyen local.

7.5 Objet géré «Software Distributor» (*distributeur de logiciel*)

Un objet géré «Software Distributor» est un objet statique représentant le ou les mécanismes de livraison d'un système géré. Il s'agit d'un objet géré qui fournit du logiciel à un système géré cible lorsqu'il reçoit une opération de livraison d'un système gestionnaire. Les paramètres de l'opération de livraison peuvent être utilisés pour indiquer l'ensemble de logiciels à livrer, la cible destinataire de la livraison et le choix du mécanisme de transfert. Quoique cette classe d'objets puisse être utilisée pour initialiser la livraison au moyen de divers mécanismes de transfert, elle ne modélise aucune de ces classes, ceci étant l'objet de spécialisations.

Cet objet géré émet une notification contenant le résultat de la distribution lorsque celle-ci est terminée.

7.5.1 Opérations sur l'objet géré «Software Distributor»

Les opérations suivantes s'appliquent à un objet géré «distributeur de logiciel»:

- Create (*création*): crée un nouvel objet géré «distributeur de logiciel».
- Deliver (*livraison*): incite l'objet géré «distributeur de logiciel» à créer (par une méthode qui est en dehors du domaine d'application de la présente Recommandation | Norme internationale) le logiciel spécifié sur le système géré cible et, par effet de bord sur ce même système, entraîne la création de toute ressource devant être associée à cet objet géré «unité de logiciel».
- Delete (*suppression*): provoque la suppression de l'objet géré «distributeur de logiciel» sur le système géré.

7.5.2 Etats de l'objet géré «Software Distributor»

Un objet géré «distributeur de logiciel» peut se trouver dans l'un des états suivants en fonction de la dernière opération qui a été effectuée sur lui. Les états possibles sont «créé» et «inexistant» (voir le Tableau 5).

Tableau 5 – Transitions d'état de l'objet géré «distributeur de logiciel»

Etat de départ/état d'arrivée	Inexistant	Créé
Inexistant	–	Créé
Créé	Supprimé	Livré

7.6 Relations

Les relations suivantes ont été identifiées entre objets gérés logiciels ainsi qu'entre objets gérés logiciels et d'autres objets gérés:

- Dependency (*dépendance*): cette relation peut être utilisée pour modéliser le fait qu'un objet logiciel géré dépend d'une certaine manière de la présence d'un autre objet logiciel géré. Une telle relation peut être utilisée pour apporter des corrections à un modèle.
- Configuration (*configuration*): cette relation peut être utilisée pour modéliser le fait qu'un objet géré «unité de logiciel» peut affecter le comportement d'un autre objet géré «unité de logiciel». Une police de caractères supplémentaire peut par exemple être modélisée par une relation de configuration. Cette relation peut également être utilisée pour apporter des améliorations ou des corrections à un modèle.
- Utilisation (*utilisation*): cette relation peut être utilisée pour indiquer quels sont les autres objets gérés qui utilisent des objets gérés logiciels. De tels objets représenteront probablement des processus en activité sur le système géré.

Une faute d'un objet géré peut avoir pour effet de provoquer une faute pour tout objet géré qui dépend de lui. Une telle faute ne provoquera toutefois pas de faute pour les objets qui lui sont liés par une relation de configuration, quoique le comportement de ces objets puisse être modifié. La détection de logiciel fautif peut entraîner comme résultat l'exigence d'installer une nouvelle copie du logiciel ou de livrer et d'installer une version mise à jour.

La spécification de l'une quelconque de ces relations est en dehors du domaine d'application de la présente Recommandation | Norme internationale compte tenu du fait que les relations dépendent des applications.

8 Définitions génériques

8.1 Objets gérés

8.1.1 Software (*logiciel*)

La classe d'objets «Software» est une classe d'objets gérés représentant une information logique stockée dans un élément géré incluant des programmes et des tables de données.

Cette classe d'objets gérés possède les caractéristiques obligatoires suivantes:

- softwareID.

La classe d'objets gérés contient les paquetages conditionnels suivants (avec les caractéristiques suivantes):

- createDeleteNotificationsPackage (notifications objectCreation et objectDeletion);
- attributeValueChangeNotificationPackage (notification attributeValueChange);
- stateChangeNotificationPackage (notification stateChange);
- administrativeOperationalStatesPackage (attributs administrativeState et operationalState);

- affectedObjectListPackage (attribut affectedObjectList);
- softwareProcessingErrorAlarmPackage [attribut alarmStatus (en lecture seulement) et notification processingErrorAlarm];
- userLabelPackage (attribut userLabel);
- vendorNamePackage (attribut vendorName);
- versionPackage (attribut version);
- currentProblemListPackage (attribut currentProblemList).

Cette classe d'objets gérés est définie dans la Recommandation M.3100.

8.1.2 Software Unit (*unité de logiciel*)

La classe d'objets «Software Unit» est une classe d'objets gérés qui fournit une information, pouvant être administrée, associée au logiciel résidant physiquement dans la mémoire. Le logiciel peut se trouver sous forme de fichier exécutable telle qu'un programme logiciel, ou sous forme de fichier non exécutable telle que des données ou une table de références croisées. L'unité de logiciel est une sous-classe de la classe objet logiciel définie dans la Recommandation M.3100.

Cette classe d'objets gérés possède les caractéristiques obligatoires suivantes:

- attribut administrativeState;
- attribut availabilityStatus;
- attribut operationalState;
- attribut proceduralStatus;
- notification processingErrorAlarm avec le paramètre alarmEffectOnService.

La classe d'objets gérés contient les paquetages conditionnels suivants (avec les caractéristiques suivantes):

- appliedPatchPackage (attribut appliedPatches);
- checkSumPackage (attribut checkSum);
- fileInformationPackage (attributs dateOfCreation, identityOfCreator, dateOfLastModification, identityOfLastModifier, dateDelivered et dateInstalled);
- filePackage (attributs fileType, fileLocation et fileSize);
- informationAutoBackupPackage (notification autoBackupReport, attributs futureAutoBackupTriggerThreshold et futureAutoBackupDestination);
- informationAutoRestorePackage (notification autoRestoreReport, attributs futureAutoRestoreSource et futureAutoRestoreAllowed);
- informationBackupPackage (action backup, attributs lastBackupTime et lastBackupDestination);
- informationRestorePackage (action restore, attributs lastRestoreTime et lastRestoreSource);
- installPackage (action install);
- noteFieldPackage (attribut noteField);
- revertPackage (action revert);
- terminateValidationPackage (action terminateValidation);

- usageStatePackage (attribut usageState);
- validationPackage (action validate).

8.1.3 Executable Software (*logiciel exécutable*)

La classe d'objets «Executable Software» est une classe d'objets gérés qui fournit une information, pouvant être administrée, associée à un programme exécutable au sein du système géré. Cette classe d'objets est une sous-classe de la classe d'objets gérés «Software Unit». Le programme exécutable lui-même, qui peut être constitué de segments de code associés ou non à des segments de données, peut être dans un format non normalisé dépendant de la machine et qui est généralement illisible pour le système gestionnaire.

Une action, dénommée executeProgram et définie plus loin, peut être utilisée pour exécuter le programme représenté par l'instance de l'objet logiciel exécutable. L'attribut usageState est utilisé pour indiquer s'il existe des exécutions du programme en cours. La valeur «repos» indique qu'il n'y a pas d'exécution en cours. La signification des valeurs «actif» et «occupé» est propre à l'implémentation et ne fait pas partie du domaine d'application de la présente Recommandation | Norme internationale. La valeur «actif» pourrait par exemple signifier qu'il existe un certain nombre d'exécutions en cours, alors que la valeur «occupé» pourrait signifier que le programme a atteint sa capacité maximale et que de nouvelles demandes d'exécution, au moyen de l'action executeProgram ou par un moyen local, seront rejetées ou mises en attente. Des spécialisations peuvent souhaiter inclure un seuil pour le nombre d'utilisateurs exigés pour atteindre la capacité maximale.

Cette classe d'objets gérés possède les caractéristiques supplémentaires obligatoires suivantes:

- attribut usageState.

Cette classe d'objets gérés contient les paquetages supplémentaires conditionnels suivants (avec les caractéristiques suivantes):

- executeProgramPackage (action executeProgram).

8.1.4 Software Distributor (*distributeur de logiciel*)

Un objet géré «Software Distributor» est un objet statique représentant un ou plusieurs mécanismes de livraison d'un système géré. Il s'agit d'un objet géré qui distribue du logiciel spécifié à un système géré cible spécifié lorsqu'il reçoit de la part du système gestionnaire une opération de livraison destinée au système géré. Les paramètres de l'opération de livraison peuvent être utilisés pour indiquer l'ensemble de logiciels à livrer, la cible destinataire de la livraison et le choix du mécanisme de transfert. Quoique cette classe d'objets puisse être utilisée pour initialiser la livraison au moyen de divers mécanismes de transfert, elle ne modélise aucune de ces classes. Les modèles correspondants sont des spécialisations.

Cet objet émet une notification contenant le résultat de la distribution lorsque celle-ci est terminée. La notification stateChangeNotification définie dans la Rec. X.721 du CCITT | ISO/CEI 10165-2 sera émise en cas de changement de la valeur de l'état administratif ou de l'état opérationnel. La classe d'objets gérés «Software Distributor» est une sous-classe de la classe sommitale d'objets gérés définie dans la Rec. X.721 du CCITT | ISO/CEI 10165-2.

Cette classe d'objets gérés possède les caractéristiques obligatoires suivantes:

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

8.2 Attributs

8.2.1 appliedPatches (*corrections appliquées*)

Cet attribut identifie les corrections qui ont été appliquées et qui restent en vigueur dans l'unité de logiciel représentée par l'instance de l'objet unité de logiciel. Les corrections sont des mises à jour du logiciel. La valeur de cet attribut est définie en lecture seulement et il est mis à jour d'une manière automatique lorsqu'une correction est effectuée sur le

logiciel. La syntaxe de cet attribut est une liste d'identificateurs de correction, un identificateur de correction pouvant être un choix entre une instance d'objet si la correction est représentée sous la forme d'un objet géré «unité de logiciel» et une chaîne de caractères graphiques dans le cas contraire.

8.2.2 checkSum (*somme de contrôle*)

Cet attribut identifie la somme de contrôle de l'information du logiciel représentée par l'instance de l'objet unité de logiciel.

8.2.3 dateDelivered (*date de livraison*)

Cet attribut identifie la date à laquelle l'information représentée par l'instance de l'objet unité de logiciel a été fournie au système géré. Les valeurs valides pour cet attribut sont un temps ASN.1 généralisé ou «Nul» si l'information n'a pas été fournie.

8.2.4 dateInstalled (*date d'installation*)

Cet attribut identifie la date à laquelle l'information représentée par l'instance de l'objet unité de logiciel a été installée. Les valeurs valides pour cet attribut sont un temps ASN.1 généralisé ou «Nul» si l'information n'a pas été fournie.

8.2.5 dateOfCreation (*date de création*)

Cet attribut identifie la date de création de l'objet géré «unité de logiciel». La syntaxe est celle d'un temps ASN.1 généralisé. Cet attribut a obligatoirement une valeur initiale égale à la date effective de création de l'instance de l'objet déterminé par un moyen local.

8.2.6 dateOfLastModification (*date de dernière modification*)

Cet attribut identifie la date de la dernière ou de la plus récente modification (par exemple de la création ou de la désinstallation) de l'information représentée par le logiciel. Les valeurs valides pour cet attribut sont un temps ASN.1 généralisé ou «Nul» si l'information n'a pas été fournie.

NOTE – La date de désinstallation devient la date de dernière modification si la dernière modification est une désinstallation.

8.2.7 fileLocation (*emplacement de fichier*)

Cet attribut identifie l'adresse physique ou logique complète du ou des fichiers de logiciel représentés par l'instance de l'objet unité de logiciel. La syntaxe de cet attribut est un ensemble de chaînes de caractères graphiques ou Nul. Le format d'une adresse dépend de l'implémentation compte tenu des conventions d'adressage du système géré considéré, de sorte que la représentation syntaxique est une chaîne de caractères graphiques. La valeur nulle de cet attribut indique que l'information à laquelle l'objet géré «unité de logiciel» s'applique n'a pas encore été fournie au système géré.

8.2.8 fileSize (*taille de fichier*)

Cet attribut indique la taille de l'objet géré «unité de logiciel».

8.2.9 fileType (*type de fichier*)

Cet attribut indique le type de l'unité de logiciel. Les types d'unités de logiciel possibles sont un fichier binaire non structuré (par exemple un fichier exécutable), un fichier texte non structuré (par exemple un fichier non exécutable), un fichier de blocs spéciaux, etc.

8.2.10 futureAutoBackupDestination (*destination de sauvegarde automatique future*)

Cet attribut spécifie la destination vers laquelle sera sauvegardée l'information représentée par l'instance de l'objet. Le critère de sauvegarde est défini par l'attribut futureAutoBackupTriggerThreshold de l'instance de l'objet. La destination peut être une autre instance d'objet de la même classe d'objet existant sur le même système géré local, un système ouvert distant accédé au moyen d'un protocole de transfert de fichier tel que FTAM, ou le système gestionnaire accédé en ligne au moyen de la notification autoBackupReport.

8.2.11 futureAutoBackupTriggerThreshold (*seuil de déclenchement de sauvegarde automatique future*)

Cet attribut spécifie le seuil de déclenchement d'une sauvegarde automatique de l'information représentée par l'instance de l'objet. Le seuil est défini par le comptage du nombre de modifications de l'information. Une sauvegarde automatique sera faite une fois ce seuil atteint. La destination de la sauvegarde est définie par l'attribut futureAutomaticBackupDestination. Les sauvegardes de ce type sont faites en plus des sauvegardes périodiques planifiées. Une notification autoBackupReport sera émise par l'objet une fois la sauvegarde effectuée.

8.2.12 futureAutoRestoreAllowed (*autorisation de restauration automatique future*)

Cet attribut spécifie si la restauration de l'information représentée par cette instance de l'objet géré est autorisée. La syntaxe de cet attribut est un type BOOLEAN ASN.1, la valeur «Vrai» signifiant une autorisation et «Faux» signifiant une interdiction. Les critères de déclenchement de la restauration automatique sont propres au système.

8.2.13 futureAutoRestoreSource (*source de restauration automatique future*)

Cet attribut spécifie la source de l'information utilisée pour restaurer l'information représentée par l'instance de l'objet géré. La source est un objet géré local ou un système distant. Les critères de déclenchement de la restauration automatique sont propres au système.

8.2.14 identityOfCreator (*identité du créateur*)

Cet attribut identifie l'entité créatrice de l'objet géré.

8.2.15 identityOfLastModifier (*identité du dernier modificateur*)

Cet attribut identifie le dernier ou le plus récent auteur d'une modification de l'information représentée par l'instance de l'objet unité de logiciel.

8.2.16 lastBackupDestination (*dernière destination de sauvegarde*)

Cet attribut identifie la destination, si elle existe, vers laquelle l'information représentée par l'instance de l'objet géré a été sauvegardée.

8.2.17 lastBackupTime (*date de la dernière sauvegarde*)

Cet attribut identifie la date de la dernière sauvegarde de l'information représentée par l'instance de l'objet géré. Les valeurs valides pour cet attribut sont un temps ASN.1 généralisé ou «Nul» si aucune sauvegarde n'a été effectuée.

8.2.18 lastRestoreSource (*source de la dernière restauration*)

Cet attribut identifie la source, si elle existe, à partir de laquelle l'information représentée par l'instance de l'objet géré a été restaurée.

8.2.19 lastRestoreTime (*date de la dernière restauration*)

Cet attribut identifie la date de la dernière restauration de l'information représentée par l'instance de l'objet géré. Les valeurs valides pour cet attribut sont un temps ASN.1 généralisé ou «Nul» si aucune restauration n'a été effectuée.

8.2.20 noteField (*champ de note*)

Cet attribut contient tout type d'information ou des commentaires associés à l'objet géré, incluant toute instruction particulière d'installation, toutes valeurs et tous paramètres initiaux, ou toute information nécessaire pour activer les fonctions de l'objet géré, etc.

8.2.21 softwareDistributorId (*identité du distributeur du logiciel*)

L'attribut softwareDistributorId est utilisé pour identifier les instances de la classe d'objets gérés «distributeur de logiciel».

8.3 Actions

L'ensemble des paramètres et de la sémantique génériques d'action définis par la présente Recommandation | Norme internationale donne le détail des paramètres généraux suivants du service M-ACTION défini par la Rec. X.710 du CCITT et l'ISO/CEI 9595:

- action type (*type d'action*);
- action information (*information de l'action*);
- action reply (*réponse à l'action*).

8.3.1 backup (sauvegarde)

Le service «backup» est utilisé par un système gestionnaire pour demander l'exécution d'une sauvegarde à destination de l'information représentée par l'instance de l'objet cible (c'est-à-dire l'objet géré représentant le logiciel qui est en train de ...).

Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre «type d'action» indiquera une sauvegarde. Le paramètre «information de l'action» indiquera la destination vers laquelle l'information sera sauvegardée. Des destinations possibles sont les suivantes:

- un objet géré local. L'opération de sauvegarde sera effectuée dans ce cas d'une manière interne au sein du système géré;
- le système gestionnaire qui émet cette action. Une copie de l'information de sauvegarde sera envoyée en ligne dans ce cas dans la réponse «réponse à l'action»;
- un système distant. L'information de sauvegarde sera transférée dans ce cas en temps différé vers le système distant par un moyen local.

Le paramètre «réponse à l'action» indiquera le résultat de la sauvegarde. Dans le cas d'une sauvegarde locale ou en temps différé, une valeur «Nul» indique la réussite de la sauvegarde. Dans le cas d'une sauvegarde en ligne, l'information de sauvegarde sera contenue dans le paramètre de réponse à l'action.

8.3.2 deliver (livraison)

Le service «deliver» est utilisé par un système gestionnaire pour demander la distribution d'un logiciel ou d'un ensemble de logiciels. L'information de livraison identifie le logiciel à distribuer. Une pluralité d'unités de logiciel peut être livrée en bloc à un système géré. Si l'exécution réussit, elle a pour résultat la copie du logiciel devant être distribué sur le système cible. Il est possible que des objets unité de logiciel soient créés au cours du processus.

Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre type d'action indiquera une livraison.

8.3.3 execute program (exécution de programme)

Le service «execute program» est utilisé par un système gestionnaire pour lancer l'exécution d'un programme représenté par un objet logiciel exécutable.

Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre «type d'action» indiquera une exécution de programme.

8.3.4 install (installation)

Le service «install» est utilisé par un système gestionnaire pour installer une unité de logiciel livrée au préalable. Une indication spécifique d'erreur operationStateMismatch (*discordance d'état d'opération*) est renvoyée si l'unité de logiciel ne peut pas être installée suite à une condition d'état non valide.

Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre «type d'action» indiquera une installation.

8.3.5 restore (restauration)

Le service «restore» est utilisé par un système gestionnaire pour demander l'exécution d'une restauration de l'information représentée par l'instance de l'objet cible. Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre «type d'action» indiquera une restauration. Le paramètre

«information de l'action» indiquera la source à partir de laquelle l'information sera restaurée. Des sources possibles sont les suivantes:

- un objet géré local. L'opération de restauration sera effectuée dans ce cas d'une manière interne au sein du système géré;
- le système gestionnaire qui émet cette action. Une copie de l'information de restauration sera envoyée en ligne dans ce cas dans la réponse à l'action;
- un système distant. L'information de restauration sera transférée dans ce cas en temps différé vers le système distant par un moyen local.

Cette action fait toujours l'objet d'une confirmation.

8.3.6 revert (désinstallation)

Le service «revert» est utilisé par un système gestionnaire, par exemple un système d'exploitation, pour donner l'instruction à un système géré d'effectuer une désinstallation dans l'application d'une correction ou d'un ensemble de corrections portant sur le logiciel représenté par l'objet géré «unité de logiciel».

Le service de désinstallation renverra automatiquement la valeur de l'attribut appliedPatches de l'instance de l'objet unité de logiciel à laquelle s'adresse le service. Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre «type d'action» indiquera une désinstallation. Le paramètre «information de l'action» indiquera la correction ou les corrections à inverser dans l'instance de l'objet unité de logiciel. Chaque identificateur de correction est un identificateur propre au système (chaîne de caractères imprimables) ou une instance d'objet unité de logiciel.

8.3.7 terminateValidation (fin de validation)

Le service «terminateValidation» est utilisé par un système gestionnaire pour mettre fin à une validation actuellement active (en cours d'exécution). Une précondition de cette opération est que l'objet qui reçoit la demande soit en cours de validation, dans le cas contraire la réponse à l'action indique qu'il n'existe pas de processus de validation à terminer.

Deux modes sont disponibles pour terminer une validation: le mode troncation et le mode annulation. Dans le mode troncation, la validation se termine et le résultat de la validation partielle est renvoyé dans le paramètre de réponse à l'action de l'opération M-ACTION terminateValidation. Dans le mode annulation, la validation se termine et le résultat de la validation partielle est rejeté.

Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595. Le paramètre «type d'action» indiquera une fin de validation. Le paramètre «information de l'action» indiquera le mode de fin, c'est-à-dire le mode annulation ou troncation. Le paramètre «réponse à l'action» indiquera le résultat de fin, c'est-à-dire une fin de validation pour le mode annulation, une fin de validation avec résultat partiel pour le mode de troncation ou le fait qu'il n'existe pas de validation à terminer.

8.3.8 validate (validation)

Le service «validate» est utilisé par un système gestionnaire pour demander l'exécution d'une validation de l'information représentée par l'instance d'objet unité de logiciel. Ce service utilise le service et les procédures M-ACTION définis dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le paramètre «type d'action» indiquera une validation. Le paramètre «information de l'action» utilisera, pour indiquer le type de validation à effectuer, un ensemble d'extensions de gestion. Le paramètre «information de l'action» indiquera le résultat de la validation. S'il est mis fin au processus de validation avant son achèvement normal au moyen de l'action M-ACTION terminateValidation, la réponse de cette opération indiquera la valeur «terminé» en utilisant un type ASN.1 Nul. Si le mode de fin de l'action M-ACTION validation est «troncation», le résultat de la validation tronquée contenant la partie de la validation déjà effectuée sera renvoyé par le paramètre «réponse à l'action» de l'action M-ACTION de validation.

8.4 Notifications

L'ensemble des paramètres et de la sémantique génériques d'action définis par la présente Recommandation | Norme internationale donne le détail des paramètres généraux suivants du service M-EVENT-REPORT tel qu'il est défini par la Rec. X.710 du CCITT et l'ISO/CEI 9595:

- event type (*type d'événement*);
- event information (*information d'événement*);
- event reply (*réponse d'événement*).

8.4.1 autoBackupReport (*compte rendu de sauvegarde automatique*)

La notification autoBackupReport est émise dans le but de rendre compte d'une sauvegarde automatique de l'information représentée par cet objet.

Les critères de sauvegarde automatique et la destination de la sauvegarde sont définis respectivement par les attributs futureAutoBackupTriggerThreshold et futureAutoBackupDestination de l'instance de cet objet.

La destination de la sauvegarde peut être locale (il s'agit d'une sauvegarde vers un autre objet de la même classe au sein du système géré local), le système gestionnaire ou en temps différé vers un système distant en utilisant un protocole de transfert de fichier tel que FTAM. Le résultat de la sauvegarde, c'est-à-dire son succès ou son échec sera rapporté par cette indication dans le cas d'une sauvegarde locale ou en temps différé. Dans le cas d'une sauvegarde vers le système gestionnaire, l'information sauvegardée sera envoyée en ligne comme partie du paramètre «information d'événement» de la notification.

Ce service utilise le service et les procédures M-EVENT-REPORT définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595. Le paramètre «type d'événement» indiquera un compte rendu de sauvegarde automatique.

8.4.2 autoRestoreReport (*compte rendu de restauration automatique*)

La notification autoRestoreReport est émise par l'objet géré lorsqu'une restauration automatique de l'information représentée par cette instance de l'objet a eu lieu. Le critère de déclenchement de la restauration automatique est propre au système. La source de l'information restaurée et le résultat de l'opération, c'est-à-dire la réussite ou l'échec, feront l'objet d'un compte rendu dans la notification.

Ce service utilise le service et les procédures M-EVENT-REPORT définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595. Le paramètre «type d'événement» indiquera un compte rendu de restauration automatique.

8.4.3 deliverResultNotification (*notification de résultat de livraison*)

Ce type de notification est utilisé pour rendre compte des résultats de livraisons. Le paramètre résultat peut prendre une des valeurs suivantes: effectué, échec ou inconnu.

Ce service utilise le service et les procédures M-EVENT-REPORT définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595. Le paramètre «type d'événement» indiquera le résultat de la livraison.

8.5 Paramètres

8.5.1 alarmEffectOnServiceParameter (*effet de l'alarme sur le service*)

Le paramètre alarmEffectOnServiceParameter est prévu pour être inclus dans le paramètre ManagementExtension (*extension de gestion*) du paramètre AdditionalInformation (*information supplémentaire*) du paramètre AlarmInfo (*information d'alarme*) du service Alarm Reporting (*compte rendu d'alarme*). Le paramètre indique si l'alarme a un effet sur le service.

Le paramètre ManagementExtension se présente sous la forme suivante (voir la Rec. X.721 du CCITT | ISO/CEI 10165-2):

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

L'identificateur du type OBJECT IDENTIFIER véhiculé par le champ «identifier» doit être la valeur sous laquelle la définition du présent paramètre est enregistrée. Le type véhiculé dans le champ «information» doit être le type identifié par l'expression WITH SYNTAX figurant dans la présente définition de paramètre.

8.5.2 softwareProcessingFailureParameter (*faute de traitement de logiciel*)

Le paramètre softwareProcessingFailureParameter définit la syntaxe des données qui doivent être renvoyées dans une réponse d'erreur CMIP ProcessingFailure (faute de traitement) à une M-ACTION si une demande d'opération sur un logiciel est refusée en raison d'erreurs autres que celles qui sont déjà définies dans le protocole CMIP-1. Les attributs de la syntaxe des données renvoyées incluent les attributs d'état de l'objet émetteur. Tout autre attribut susceptible de s'appliquer peut être présent, mais ceci est un problème local.

8.6 Corrélations de nom

8.6.1 Sous-système – Software Distributor (*distributeur de logiciel*)

Cette corrélation de nom est utilisée pour donner un nom à un objet distributeur de logiciel relativement à un objet sous-système.

8.6.2 Système – Software Distributor

Cette corrélation de nom est utilisée pour donner un nom à un objet distributeur de logiciel relativement à un objet système.

8.6.3 Sous-système – Software Unit (*unité de logiciel*)

Cette corrélation de nom est utilisée pour donner un nom à un objet unité de logiciel relativement à un objet sous-système.

8.6.4 Système – Software Unit

Cette corrélation de nom est utilisée pour donner un nom à un objet unité de logiciel relativement à un objet système.

9 Définition de services

9.1 Introduction

Le présent article définit des services ayant pour objet de livrer des logiciels, d'exécuter des programmes, d'installer et de désinstaller des logiciels, de terminer la validation de logiciels et de valider des logiciels. Elle définit également des services de compte rendu de sauvegarde et de restauration automatiques de logiciel. Elle définit en outre des services génériques de sauvegarde et de restauration.

9.2 Service Backup (*sauvegarde*)

Le service de sauvegarde permet à un gestionnaire de demander à un autre système ouvert (le système géré) de sauvegarder une entité. Le Tableau 6 donne la liste des paramètres de ce service.

Le service de sauvegarde utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre réponse de sauvegarde figurera dans une réponse positive, le paramètre erreurs sera présent dans le cas contraire.

9.3 Service Deliver (*livraison*)

Le service de livraison permet à un gestionnaire de demander à un autre système ouvert (le système géré) de démarrer une livraison de logiciel. Le Tableau 7 donne la liste des paramètres de ce service.

Le service de livraison utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre information de réponse de livraison figurera dans une réponse positive, le paramètre erreurs sera présent dans le cas contraire.

9.4 Service Execute program (*exécution de programme*)

Le service d'exécution de programme permet à un gestionnaire de demander à un autre système ouvert (le système géré) d'exécuter un programme. Le Tableau 8 donne la liste des paramètres de ce service.

Le service d'exécution de programme utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre réponse d'exécution de programme figurera dans une réponse positive, le paramètre erreurs sera présent dans le cas contraire.

Tableau 6 – Paramètres du service «Backup»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type de sauvegarde	M	C(=)
Argument de sauvegarde	M	–
Destination de la sauvegarde	M	–
Information supplémentaire	U	–
Date courante	–	P
Réponse de sauvegarde	–	C
Réponse	–	C
Information supplémentaire	–	U
Erreurs	–	C

Tableau 7 – Paramètres du service «Deliver»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type de livraison	M	C(=)
Information de livraison	M	–
Identificateur de livraison	U	–
Logiciel cible	M	–
Système cible	U	–
Information de transfert	U	–
Information supplémentaire	U	–
Date courante	–	P
Info. de réponse de livraison	–	C
Identificateur de livraison	–	U
Résultat de livraison	–	C
Information supplémentaire	–	U
Erreurs	–	C

Tableau 8 – Paramètres du service «Execute program»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type d'exécution de programme	M	C(=)
Info. d'exécution de programme	M	–
Date courante	–	P
Réponse d'exéc. de programme	–	C
Identificateur de processus	–	C
Propriétaire du processus	–	C
Date de début	–	C
Information supplémentaire	–	U
Erreurs	–	C

9.5 Service Install (*installation*)

Le service d'installation permet à un gestionnaire de demander à un autre système ouvert (le système géré) d'installer du logiciel. Le Tableau 9 donne la liste des paramètres de ce service.

Le service d'installation utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre erreurs sera présent dans la réponse si celle-ci n'est pas positive.

Tableau 9 – Paramètres du service «Install»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type d'installation	M	C(=)
Information d'installation	M	–
Logiciel cible	M	–
Information d'installation	M	–
Date courante	–	P
Erreurs	–	C

9.6 Service Restore (*restauration*)

Le service de restauration permet à un gestionnaire de demander à un autre système ouvert (le système géré) de restaurer une entité. Le Tableau 10 donne la liste des paramètres de ce service.

Le service de restauration utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre erreurs sera présent dans la réponse si celle-ci n'est pas positive.

Tableau 10 – Paramètres du service «Restore»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	—	P
Mode	P	—
Classe d'objets de base	P	—
Instance d'objet de base	P	—
Domaine d'application	P	—
Filtre	P	—
Classe d'objets gérés	—	P
Instance d'objet géré	—	P
Contrôle d'accès	P	—
Synchronisation	P	—
Type de restauration	M	C(=)
Argument de restauration	M	—
Source de restauration	M	—
Information supplémentaire	U	—
Date courante	—	P
Erreurs	—	C

9.7 Service Revert (*désinstallation*)

Le service de désinstallation permet à un gestionnaire de demander à un autre système ouvert (le système géré) de désinstaller du logiciel. Le Tableau 11 donne la liste des paramètres de ce service.

Le service de désinstallation utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre réponse de désinstallation figurera dans une réponse positive, le paramètre erreurs sera présent dans le cas contraire.

9.8 Service Terminate validation (*fin de validation*)

Le service de fin de validation permet à un gestionnaire de demander à un autre système ouvert (le système géré) de mettre fin à une validation de logiciel. Le Tableau 12 donne la liste des paramètres de ce service.

Le service de fin de validation utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre réponse de fin de validation figurera dans une réponse positive, le paramètre erreurs sera présent dans le cas contraire.

Tableau 11 – Paramètres du service «Revert»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type de désinstallation	M	C(=)
Information de désinstallation	M	–
Identificateur de correction	M	–
Information supplémentaire	U	–
Date courante	–	P
Réponse de désinstallation	–	C
Corrections appliquées	–	C
Information supplémentaire	–	U
Erreurs	–	C

Tableau 12 – Paramètres du service «Terminate validation»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type de fin de validation	M	C(=)
Argument de fin de validation	M	–
Information de fin de validation	M	–
Information supplémentaire	U	–
Date courante	–	P
Réponse de fin de validation	–	C
Erreurs	–	C

9.9 Service Validate (*validation*)

Le service de validation permet à un gestionnaire de demander à un autre système ouvert (le système géré) de valider du logiciel. Le Tableau 13 donne la liste des paramètres de ce service.

Le service de validation utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Le paramètre réponse de validation figurera dans une réponse positive, le paramètre erreurs sera présent dans le cas contraire.

9.10 Service Auto backup report (*compte rendu de sauvegarde automatique*)

Le service de compte rendu de sauvegarde automatique permet à un système ouvert (le système géré) de rendre compte d'une sauvegarde automatique. Le Tableau 14 donne la liste des paramètres de ce service.

Le service de compte rendu de sauvegarde automatique utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-EVENT-REPORT définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

9.11 Service Auto restore report (*compte rendu de restauration automatique*)

Le service de compte rendu de restauration automatique permet à un système ouvert (le système géré) de rendre compte d'une restauration automatique. Le Tableau 15 donne la liste des paramètres de ce service.

Le service de compte rendu de restauration automatique utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-EVENT-REPORT définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Tableau 13 – Paramètres du service «Validate»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Identificateur chaîné	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type de validation	M	C(=)
Information de validation	M	–
Date courante	–	P
Réponse de validation	–	C
Erreurs	–	C

Tableau 14 – Paramètres du service «Auto backup report»

Nom du paramètre	Demande/indication	Réponse/confirmation
Identificateur de l'invocation	P	P
Mode	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Type de compte rendu de sauvegarde automatique	M	C(=)
Date de l'événement	P	–
Information de compte rendu de sauvegarde automatique	M	–
Résultat de la sauvegarde	M	–
Information supplémentaire	U	–
Date courante	–	P
Réponse d'événement	–	–
Erreurs	–	C

Tableau 15 – Paramètres du service «Auto restore report»

Nom du paramètre	demande/indication	réponse/confirmation
Identificateur de l'invocation	P	P
Mode	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Type de compte rendu de restauration automatique	M	C(=)
Date de l'événement	P	–
Information de compte rendu de restauration automatique	M	–
Source	M	–
Réussite	M	–
Information supplémentaire	U	–
Date courante	–	P
Réponse d'événement	–	–
Erreurs	–	C

9.12 Service Deliver result notification (*notification de résultat de livraison*)

Le service de notification de résultat de livraison permet à un système ouvert (le système géré) de rendre compte du résultat d'une livraison de logiciel. Le Tableau 16 donne la liste des paramètres de ce service.

Le service de notification de résultat de livraison utilise les paramètres définis dans l'article 8 en plus des paramètres du service M-EVENT-REPORT définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595.

Tableau 16 – Paramètres du service «Deliver result notification»

Nom du paramètre	demande/ indication	réponse/ confirmation
Identificateur de l'invocation	P	P
Mode	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Type de notification de résultat de livraison	M	C(=)
Date de l'événement	P	–
Information de résultat de livraison	M	–
Identificateur de livraison	U	–
Résultat de livraison	M	–
Information supplémentaire	U	–
Date courante	–	P
Réponse d'événement	–	–
Erreurs	–	C

10 Unités fonctionnelles

La présente Recommandation | Norme internationale définit trois unités fonctionnelles pour la gestion de logiciel et d'autres entités:

- a) l'unité fonctionnelle de supervision de logiciel;
- b) l'unité fonctionnelle de livraison de logiciel;
- c) l'unité fonctionnelle de sauvegarde et de restauration.

L'unité fonctionnelle de supervision de logiciel nécessite la prise en charge des services exécution de programme, installation, désinstallation, fin de validation et validation. L'unité fonctionnelle de livraison de logiciel nécessite la prise en charge des services de livraison et de notification de résultat de livraison. L'unité fonctionnelle de sauvegarde et de restauration nécessite la prise en charge des services de sauvegarde et de restauration, de notification de sauvegarde automatique et de notification de restauration automatique.

11 Protocole

11.1 Eléments de procédure

11.1.1 Procédure de sauvegarde

11.1.1.1 Rôle du gestionnaire

11.1.1.1.1 Invocation

Les procédures de sauvegarde sont lancées par la primitive de sauvegarde. Lorsqu'elle reçoit une primitive de sauvegarde, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive de sauvegarde. Le mode avec confirmation sera utilisé.

11.1.1.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération de sauvegarde, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de sauvegarde une primitive de confirmation de sauvegarde dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure de sauvegarde.

NOTE – La machine SMAPM ignorerà toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de sauvegarde peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.1.2 Rôle de l'agent

11.1.1.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service de sauvegarde, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication de sauvegarde dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.1.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse de sauvegarde, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse de sauvegarde.

11.1.2 Procédure de livraison

11.1.2.1 Rôle du gestionnaire

11.1.2.1.1 Invocation

Les procédures de livraison sont lancées par la primitive de livraison. Lorsqu'elle reçoit une primitive de livraison, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive de livraison. Le mode avec confirmation sera utilisé.

11.1.2.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération de livraison, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de livraison une primitive de confirmation de livraison dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure de livraison.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de livraison peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.2.2 Rôle de l'agent

11.1.2.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service de livraison, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication de livraison dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.2.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse de livraison, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse de livraison.

11.1.3 Procédure d'exécution de programme

11.1.3.1 Rôle du gestionnaire

11.1.3.1.1 Invocation

Les procédures d'exécution de programme sont lancées par la primitive d'exécution de programme. Lorsqu'elle reçoit une primitive d'exécution de programme, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive d'exécution de programme. Le mode avec confirmation sera utilisé.

11.1.3.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération d'exécution de programme, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service d'exécution de programme une primitive de confirmation d'exécution de programme dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure d'exécution de programme.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service d'exécution de programme peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.3.2 Rôle de l'agent

11.1.3.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service d'exécution de programme, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication d'exécution de programme dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.3.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse d'exécution de programme, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse d'exécution de programme.

11.1.4 Procédure d'installation

11.1.4.1 Rôle du gestionnaire

11.1.4.1.1 Invocation

Les procédures d'installation sont lancées par la primitive d'installation. Lorsqu'elle reçoit une primitive d'installation, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive d'installation. Le mode avec confirmation sera utilisé.

11.1.4.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération d'installation, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service d'installation une primitive de confirmation d'installation dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure d'installation.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service d'installation peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.4.2 Rôle de l'agent

11.1.4.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service d'installation, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication d'installation dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.4.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse d'installation, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse d'installation.

11.1.5 Procédure de restauration

11.1.5.1 Rôle du gestionnaire

11.1.5.1.1 Invocation

Les procédures de restauration sont lancées par la primitive de restauration. Lorsqu'elle reçoit une primitive de restauration, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive de restauration. Le mode avec confirmation sera utilisé.

11.1.5.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération de restauration, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de restauration une primitive de confirmation de restauration dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure de restauration.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de restauration peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.5.2 Rôle de l'agent

11.1.5.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service de restauration, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication de restauration dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.5.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse de restauration, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse de restauration.

11.1.6 Procédure de désinstallation

11.1.6.1 Rôle du gestionnaire

11.1.6.1.1 Invocation

Les procédures de désinstallation sont lancées par la primitive de désinstallation. Lorsqu'elle reçoit une primitive de désinstallation, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive de désinstallation. Le mode avec confirmation sera utilisé.

11.1.6.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération de désinstallation, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de désinstallation une primitive de confirmation de désinstallation dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure de désinstallation.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de désinstallation peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.6.2 Rôle de l'agent

11.1.6.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service de désinstallation, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication de désinstallation dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.6.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse de désinstallation, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse de désinstallation.

11.1.7 Procédure de fin de validation

11.1.7.1 Rôle du gestionnaire

11.1.7.1.1 Invocation

Les procédures de fin de validation sont lancées par la primitive de fin de validation. Lorsqu'elle reçoit une primitive de fin de validation, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive de fin de validation. Le mode avec confirmation sera utilisé.

11.1.7.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération de fin de validation, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de fin de validation une primitive de confirmation de fin de validation dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure de fin de validation.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de fin de validation peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.7.2 Rôle de l'agent

11.1.7.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service de fin de validation, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication de fin de validation dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.7.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse de fin de validation, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse de fin de validation.

11.1.8 Procédure de validation

11.1.8.1 Rôle du gestionnaire

11.1.8.1.1 Invocation

Les procédures de validation sont lancées par la primitive de validation. Lorsqu'elle reçoit une primitive de demande de validation, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-ACTION du service CMIS dont les paramètres seront déduits de la primitive de validation. Le mode avec confirmation sera utilisé.

11.1.8.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une opération de validation, une primitive de confirmation M-ACTION du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de validation une primitive de confirmation de validation dont les paramètres seront déduits de la primitive de confirmation M-ACTION du service CMIS, ce qui achèvera la procédure de validation.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de validation peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.8.2 Rôle de l'agent

11.1.8.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-ACTION du service CMIS contenant une unité MAPDU qui demande un service de validation, la machine SMAPM émettra à destination de l'utilisateur de la sauvegarde, si la primitive est formée correctement, une primitive d'indication de validation dont les paramètres seront dérivés de la primitive d'indication M-ACTION du service CMIS. Dans le cas contraire, la machine SMAPM construira une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-ACTION du service CMIS contenant un paramètre d'erreur.

11.1.8.2.2 Réponse

La machine SMAPM acceptera une primitive de réponse de validation, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-ACTION du service CMIS contenant des paramètres déduits de la primitive de réponse de validation.

11.1.9 Procédure de compte rendu de sauvegarde automatique

11.1.9.1 Rôle du gestionnaire

11.1.9.1.1 Invocation

Les procédures de compte rendu de sauvegarde automatique sont lancées par la primitive de demande de compte rendu de sauvegarde automatique. Lorsqu'elle reçoit une primitive de compte rendu de sauvegarde automatique, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-EVENT-REPORT du service CMIS dont les paramètres seront déduits de la primitive de compte rendu de sauvegarde automatique. La procédure du 11.1.9.1.2 ne s'applique pas dans le cas du mode sans confirmation.

11.1.9.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une notification de compte rendu de sauvegarde automatique, une primitive de confirmation M-EVENT-REPORT du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de compte rendu de sauvegarde automatique une primitive de confirmation de compte rendu de sauvegarde automatique dont les paramètres seront déduits de la primitive de confirmation M-EVENT-REPORT du service CMIS, ce qui achèvera la procédure de compte rendu de sauvegarde automatique.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de compte rendu de sauvegarde automatique peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.9.2 Rôle de l'agent

11.1.9.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-EVENT-REPORT du service CMIS contenant une unité MAPDU qui demande un service de compte rendu de sauvegarde automatique, la machine SMAPM émettra à destination de l'utilisateur du service de compte rendu de sauvegarde automatique, si la primitive est formée correctement, une primitive d'indication de compte rendu de sauvegarde automatique dont les paramètres seront dérivés de la primitive d'indication M-EVENT-REPORT du service CMIS. Dans le cas contraire, la machine SMAPM construira, dans le mode avec confirmation, une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-EVENT-REPORT du service CMIS contenant un paramètre d'erreur. La procédure du 11.1.9.2.2 ne s'applique pas dans le cas du mode sans confirmation.

11.1.9.2.2 Réponse

Dans le mode avec confirmation, la machine SMAPM acceptera une primitive de réponse de compte rendu de sauvegarde automatique, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-EVENT-REPORT du service CMIS contenant des paramètres déduits de la primitive de réponse de compte rendu de sauvegarde automatique.

11.1.10 Procédure de compte rendu de restauration automatique

11.1.10.1 Rôle du gestionnaire

11.1.10.1.1 Invocation

Les procédures de compte rendu de restauration automatique sont lancées par la primitive de demande de compte rendu de restauration automatique. Lorsqu'elle reçoit une primitive de compte rendu de restauration automatique, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-EVENT-REPORT du service CMIS dont les paramètres seront déduits de la primitive de compte rendu de restauration automatique. La procédure du 11.1.10.1.2 ne s'applique pas dans le cas du mode sans confirmation.

11.1.10.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une notification de compte rendu de restauration automatique, une primitive de confirmation M-EVENT-REPORT du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de compte rendu de restauration automatique une primitive de confirmation de compte rendu de restauration automatique dont les paramètres seront déduits de la primitive de confirmation M-EVENT-REPORT du service CMIS, ce qui achèvera la procédure de compte rendu de restauration automatique.

NOTE – La machine SMAPM ignorera toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de compte rendu de restauration automatique peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.10.2 Rôle de l'agent

11.1.10.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-EVENT-REPORT du service CMIS contenant une unité MAPDU qui demande un service de compte rendu de restauration automatique, la machine SMAPM émettra à destination de l'utilisateur du service de compte rendu de restauration automatique, si la primitive est formée correctement, une primitive d'indication de compte rendu de restauration automatique dont les paramètres seront dérivés de la primitive d'indication M-EVENT-REPORT du service CMIS. Dans le cas contraire, la machine SMAPM construira, dans le mode avec confirmation, une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-EVENT-REPORT du service CMIS contenant un paramètre d'erreur. La procédure du 11.1.10.2.2 ne s'applique pas dans le cas du mode sans confirmation.

11.1.10.2.2 Réponse

Dans le mode avec confirmation, la machine SMAPM acceptera une primitive de réponse de compte rendu de restauration automatique, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-EVENT-REPORT du service CMIS contenant des paramètres déduits de la primitive de réponse de compte rendu de restauration automatique.

11.1.11 Procédure de livraison de notification de résultat

11.1.11.1 Rôle du gestionnaire

11.1.11.1.1 Invocation

Les procédures de livraison de notification de résultat sont lancées par la primitive demande de livraison de notification de résultat. Lorsqu'elle reçoit une primitive de demande de livraison de notification de résultat, la machine SMAPM construira une unité MAPDU et émettra une primitive de demande M-EVENT-REPORT du service CMIS dont les paramètres seront déduits de la primitive de demande de livraison de notification de résultat. La procédure du 11.1.11.1.2 ne s'applique pas dans le cas du mode sans confirmation.

11.1.11.1.2 Réception de réponse

Lorsqu'elle reçoit, en réponse à une notification de livraison de résultat, une primitive de confirmation M-EVENT-REPORT du service CMIS contenant une unité MAPDU, la machine SMAPM enverra à l'utilisateur du service de livraison de notification de résultat une primitive de confirmation de livraison de notification de résultat dont les paramètres seront déduits de la primitive de confirmation M-EVENT-REPORT du service CMIS, ce qui achèvera la procédure de livraison de notification de résultat.

NOTE – La machine SMAPM ignorerà toutes les erreurs dans l'unité MAPDU reçue. L'utilisateur du service de livraison de notification de résultat peut ignorer ces erreurs ou abandonner l'association en raison de ces erreurs.

11.1.11.2 Rôle de l'agent

11.1.11.2.1 Réception de demande

Lorsqu'elle reçoit une primitive d'indication M-EVENT-REPORT du service CMIS contenant une unité MAPDU qui demande un service de livraison de notification de résultat, la machine SMAPM émettra à destination de l'utilisateur du service de livraison de notification de résultat, si la primitive est formée correctement, une primitive d'indication de livraison de notification de résultat dont les paramètres seront dérivés de la primitive d'indication M-EVENT-REPORT du service CMIS. Dans le cas contraire, la machine SMAPM construira, dans le mode avec confirmation, une unité MAPDU appropriée indiquant l'erreur et émettra une primitive de réponse M-EVENT-REPORT du service CMIS contenant un paramètre d'erreur. La procédure du 11.1.11.2.2 ne s'applique pas dans le cas du mode sans confirmation.

11.1.11.2.2 Réponse

Dans le mode avec confirmation, la machine SMAPM acceptera une primitive de réponse de livraison de notification de résultat, construira une unité MAPDU confirmant l'opération et enverra une primitive de réponse M-EVENT-REPORT du service CMIS contenant des paramètres déduits de la primitive de livraison de notification de résultat.

11.2 Syntaxe abstraite

11.2.1 Objets

La présente Recommandation | Norme internationale fait référence aux objets supports suivants dont la valeur en notation ASN.1 est spécifiée dans l'Annexe A:

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

11.2.2 Paquetages

La présente Recommandation | Norme internationale fait référence aux définitions de paquetage suivantes dont la valeur en notation ASN.1 est spécifiée dans l'Annexe 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 Attributs

La présente Recommandation | Norme internationale fait référence aux attributs de gestion spécifiques suivants dont la syntaxe abstraite est spécifiée dans l'Annexe 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

La présente Recommandation | Norme internationale fait référence aux types de notification suivants dont la syntaxe abstraite est spécifiée dans l'Annexe A:

- a) autoBackupReport;
- b) autoRestoreReport;
- c) deliverResultNotification.

11.2.5 Actions

La présente Recommandation | Norme internationale fait référence aux types spécifiques d'action suivants dont la syntaxe abstraite est spécifiée dans l'Annexe A:

- a) backup;
- b) deliver;
- c) executeProgram;
- d) install;

- e) restore;
- f) revert;
- g) terminateValidation;
- h) validate.

11.2.6 Corrélations de noms

La présente Recommandation | Norme internationale fait référence aux corrélations de noms spécifiques suivantes dont la syntaxe abstraite est spécifiée dans l'Annexe A:

- a) softwareDistributor-subsystem;
- b) softwareDistributor-system;
- c) softwareUnit-subsystem;
- d) softwareUnit-system.

11.3 Négociation des unités fonctionnelles

La présente Recommandation | Norme internationale assigne l'identificateur suivant:

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

comme valeur pour le type ASN.1 functionalUnitPackageId défini dans la Rec. X.701 du CCITT | ISO/CEI 10040 utilisé dans la négociation des unités fonctionnelles suivantes:

- 0 unité fonctionnelle de supervision de logiciel
- 1 unité fonctionnelle de livraison de logiciel
- 2 unité fonctionnelle de sauvegarde et de restauration

le nombre identifie la position de bit assignée à l'unité fonctionnelle et le nom donnant la référence de l'unité fonctionnelle définie dans l'article 10.

Le mécanisme de négociation de l'unité de compte rendu d'alarme dans le contexte de l'application de gestion-systèmes est défini par la Rec. X.701 du CCITT | ISO/CEI 10040.

NOTE – La prescription de négociation des unités fonctionnelles est spécifiée par le contexte de l'application.

12 Relations avec d'autres fonctions

Les fonctions suivantes sont fournies par d'autres fonctions de gestion-systèmes:

- performance du logiciel, couverte par la fonction de récapitulation (voir la Rec. UIT-T X.738 | ISO/CEI 10164-13) et la fonction d'objets et attributs de mesure (voir la Rec. UIT-T X.739 | ISO/CEI 10164-11);
- trace d'audit du logiciel, couverte par la fonction de trace d'audit de sécurité (voir la Rec. UIT-T X.740 | ISO/CEI 10164-8);
- prise en charge de la sécurité du logiciel, couverte par les objets et les attributs pour le contrôle d'accès (voir la Rec. UIT-T X.741 | ISO/CEI 10164-9);
- comptabilisation de l'utilisation du logiciel, couverte par la fonction de mesure d'utilisation (voir la Rec. UIT-T X.742 | ISO/CEI 10164-10);
- tests de logiciel, comprenant l'installation de l'environnement de test, l'exécution du test de logiciel, la mise en place de point d'arrêt, la suspension et la fermeture de l'environnement de test, couverts par la fonction de gestion de test (voir la Rec. UIT-T X.745 | ISO/CEI 10164-12);
- ordonnancement des fonctions et de l'exploitation du logiciel, couvert par la fonction d'ordonnancement (voir la Rec. UIT-T X.746 | ISO/CEI 10164-15).

13 Conformité

Les implémentations déclarant la conformité à la présente Recommandation | Norme internationale satisferont aux prescriptions de conformité définies dans les paragraphes suivants.

13.1 Conformité statique

L'implémentation se conformera aux prescriptions de la présente Recommandation | Norme internationale dans le rôle de gestionnaire, dans le rôle d'agent ou dans les deux rôles. Une déclaration de conformité à l'un au moins de ces rôles doit être faite dans le Tableau B.1.

Si une déclaration de conformité est faite pour la prise en charge dans le rôle de gestionnaire, l'implémentation prendra en charge, pour tout objet géré ou sous-classe d'objets gérés définis dans la présente Recommandation | Norme internationale, au moins une opération de gestion sur un tel objet/sous-classe ou au moins une notification issue d'un tel objet/sous-classe. Les prescriptions de conformité dans le rôle de gestionnaire pour ces opérations et notifications de gestion sont spécifiées dans l'Annexe B et d'autres tableaux référencés par l'Annexe B.

Si une déclaration de conformité est faite pour la prise en charge dans le rôle d'agent, l'implémentation prendra en charge une ou plusieurs instances des classes d'objets gérés ou de leurs sous-classes dont la spécification est donnée dans le Tableau B.4. La conformité dans le rôle d'agent pourra également être déclarée pour des sous-classes de l'objet «log record».

L'implémentation supportera la syntaxe de transfert dérivée des règles de codage spécifiées dans la Rec. X.209 du CCITT | ISO/CEI 8825 dénommée {joint-iso-ccitt asn1(1) basicEncoding(1)} pour les types abstraits de données référencés par les définitions pour lesquelles la prise en charge est déclarée.

13.2 Conformité dynamique

Des implémentations déclarant la conformité à la présente Recommandation | Norme internationale prendront en charge les éléments de procédure et les déclarations de sémantique correspondant aux définitions pour lesquelles la prise en charge est déclarée.

13.3 Prescriptions concernant les déclarations de conformité d'implémentation de gestion

Tout formulaire de déclaration MCS, formulaire de déclaration MICS, formulaire de déclaration MOCS, formulaire de déclaration MRCS qui se conforme à la présente Recommandation | Norme internationale sera techniquement identique aux formulaires spécifiés dans les Annexes B, C, D et E en conservant la numérotation des tables et la numérotation des index des items, les différences portant uniquement sur la pagination et les en-têtes de page.

Le fournisseur d'une implémentation déclarant la conformité à la présente Recommandation | Norme internationale devra remplir, comme partie des prescriptions de conformité, une copie de récapitulatif de conformité de gestion (MCS) fourni dans l'Annexe B ainsi que tout autre formulaire de déclaration ICS pour lequel le récapitulatif MCS indique qu'il s'applique. Une déclaration ICS se conformant à la présente Recommandation | Norme internationale devra:

- décrire une implémentation se conformant à la présente Recommandation | Norme internationale;
- avoir été réalisée conformément aux instructions de réalisation données dans la Rec. UIT-T X.724 | ISO/CEI 10165-6;
- comporter l'information nécessaire pour identifier sans ambiguïté le fournisseur et la mise en œuvre.

Des déclarations de conformité à l'information de gestion définie dans la présente Recommandation | Norme internationale dans des classes d'objets gérés définies ailleurs incluront les prescriptions du formulaire MIDS, tel qu'il est spécifié dans l'Annexe F, dans la déclaration MOCS de la classe d'objets gérés.

Annexe A

Définition de l'information de gestion

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

--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(8)};

--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 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.";

--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 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 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-itu-t(2) association-control(2) modules(0) apdus(0) version1(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 {

```
    distributedSoftwareId 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* --

Annexe B³⁾**Formulaire MCS**

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

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⁾ **Droits de reproduction du formulaire MCS**

Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MCS de la présente annexe pour utiliser celui-ci conformément à son objet. Ils sont également autorisés à publier le formulaire une fois celui-ci complété. Les instructions pour le formulaire MCS sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 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

Index	A	B	C	D	Support		
					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:

Index	A	B	C	D	E	F	G	Support			H	I	J	K	L
								Subindex							
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 interpreted 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

Index	Systems management functional unit name	Manager		Agent		Additional information
		Status	Support	Status	Support	
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 –.								

Annexe C⁴⁾**Formulaire MICS**

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

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) attribute(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

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	
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⁾ **Droits de reproduction du formulaire MICS**

Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MICS de la présente annexe pour utiliser celui-ci conformément à son objet. Ils sont également autorisés à publier le formulaire une fois celui-ci complété.

Table C.1 (continued) – 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	
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	fileType	{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	futureAutoRestoreAllocated	{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	nameBinding	{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

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	
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		
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	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							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	instanceDefaultValidationType	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

Index	Notifi-cation type template label	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Sup-port	Addi-tional infor-mation
		Con-straints and values	Sta-tus								
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

Index	Notification type template label	Value of object identifier for notification type	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
			Con-strain-tions and values	Sta-tus								
						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	Notifi- cation type template label	Support						Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Sup- port	Additional information
		Con- straints and values	Sta- tus	Con- firm- ed	Non- con- firm- ed	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	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-con-firm-ed								
5	objectDeletion	{dmi-not7}	o.1					4.1.4.2.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o.9		
								4.1.4.2.3	localDistingui shedName	-	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	additionalInfo rmation	{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		
								5.1	ObjectInfo		Information Syntax SEQUENCE	c:m		
								5.1.1	sourceIndicat or	{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	notificationId entifier	{dmi-att 16}	INTEGER	c:o		
								5.1.4	correlatedNoti fications	{dmi-att 12}	SET OF SEQUENCE	c:o		
								5.1.4.1	correlatedNoti fications	{dmi-att 12}	SET OF INTEGER	c:m		

Table C.3 (continued) – Notification support summary

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-con-firm-ed							
6	processi ngError Alarm	{dmi-not10}	o.1		alarmE ffectOn Service Parame ter	6.1	AlarmInfo		Information Syntax SEQUENCE	c:m			

Table C.3 (continued) – Notification support summary

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Addi-tional infor-mation
					Con-firm-ed	Non-con-firm-ed							
							6.1.4	backedUpStat us	{dmi-att 11}	BOOLEAN	o		
							6.1.5	backUpObject	{dmi-att 40}	CHOICE	o		
							6.1.5.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 14		
							6.1.5.3	localDistingui shedName	-	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	trendIndicati on	{dmi-att 30}	ENUMERATED	o		
							6.1.7	thresholdInfo	{dmi-att 29}	SEQUENCE	o		
							6.1.7.1	triggeredThre shold	-	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	observedVal ue	-	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	thresholdLev el	-	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	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Addi-tional informa-tion
					Con-firm-ed	Non-con-firm-ed								
							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	notificationId entifier	{dmi-att 16}	INTEGER	o			
							6.1.9	correlatedNoti fication	{dmi-att 12}	SET OF SEQUENCE	o			
							6.1.9.1	correlatedNoti fication	{dmi-att 12}	SET OF INTEGER	c:m			
							6.1.9.2	sourceObjectI nst	-	CHOICE	c:o			
							6.1.9.2.1	distinguished Name	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 22			
							6.1.9.2.3	localDistingui shedName	-	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	stateChangeD efinition	{dmi-att 28}	SET OF SEQUENCE	o			

Table C.3 (continued) – Notification support summary

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-con-firm-ed								
							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	oldAttributeV alue	-	ANY DEFINED BY attributeID	c:o			
							6.1.10.3	newAttribut eValue	-	ANY DEFINED BY attributeID	c:m			
							6.1.11	monitoredAttr ibutes	{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	proposedRepa irActions	{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	additionalInfo rmation	{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	stateCha-nge	{dmi-not14}	o.1				7.1	StateChangeI nfo		Information Syntax SEQUENCE	c:m			
							7.1.1	sourceIndicat or	{dmi-att 26}	ENUMERATED	c:o			
							7.1.2	attributeIdentifi erList	{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

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-con-firm-ed								
							7.1.2.2	localForm	-	INTEGER	c:o. 26			
							7.1.3	stateChangeD efinition	{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	oldAttributeV alue	-	ANY DEFINED BY attributeID	c:o			
							7.1.3.3	newAttribut eValue	-	ANY DEFINED BY attributeID	c:m			
							7.1.4	notificationId entifier	{dmi-att 16}	INTEGER	c:o			
							7.1.5	correlatedNoti fications	{dmi-att 12}	SET OF SEQUENCE	c:o			
							7.1.5.1	correlatedNoti fications	{dmi-att 12}	SET OF INTEGER	c:m			
							7.1.5.2	sourceObjectI nst	-	CHOICE	c:o			
							7.1.5.2.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 28			
							7.1.5.2.3	localDistingui shedName	-	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	additionalInfo rmation	{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	Con- straints and values	Sta- tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta- tus	Sup- port	Additional information
					Con- firm- ed	Non- con- firm- ed								
8	deliverResultNotification	{swmf-not 3}	o.1					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		
								8.1	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		

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		

Annexe D⁵⁾**Formulaire MOCS**

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

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⁾ **Droits de reproduction du formulaire MOCS**

Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MOCS de la présente annexe pour utiliser celui-ci conformément à son objet. Ils sont également autorisés à publier le formulaire une fois celui-ci complété. Les instructions pour compléter le formulaire MOCS sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 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		Additional information
				Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	
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		
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	fileType	{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

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	
29	operationalState	{dmi-att 35}	ENUMERATED	x	m	x	-	x	-	x	-	x	-	x	-	
30	packages	{dmi-att 66}	SET OF OBJECT IDENTIFIER	o	m	x	x	x	x	x	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}	ENUMERATED	-	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	-	-	-	-	-	-	-	-	-	
<p>c3: if D.4.1/1b then x else -.</p> <p>c4: if D.4.3/2a and D.4.1/1b then x else -.</p> <p>c4a: if D.4.3/2a then m else -.</p> <p>c5: if D.4.3/21a and D.4.1/1b then x else -.</p> <p>c5a: if D.4.3/21a then m else -.</p> <p>c6: if D.4.3/3a then o else -.</p> <p>c7: if D.4.3/3a then m else -.</p> <p>c8: if D.4.3/4a and D.4.1/1b then x else -.</p> <p>c9: if D.4.3/4a then m else -.</p> <p>c10: if D.4.3/6a and D.4.1/1b then x else -.</p> <p>c11: if D.4.3/6a then m else -.</p> <p>c12: if D.4.3/8a and D.4.1/1b then x else -.</p> <p>c13: if D.4.3/8a then m else -.</p> <p>c14: if D.4.3/9a and D.4.1/1b then x else -.</p> <p>c15: if D.4.3/9a then m else -.</p> <p>c16: if D.4.3/9a and D.4.1/1b then x else -.</p> <p>c17: if D.4.3/9a then m else -.</p> <p>c18: if D.4.3/9a and D.4.1/1b then x else -.</p> <p>c19: if D.4.3/9a then m else -.</p> <p>c20: if D.4.3/9a and D.4.1/1b then x else -.</p> <p>c21: if D.4.3/9a then m else -.</p> <p>c22: if D.4.3/10a and D.4.1/1b then x else -.</p> <p>c23: if D.4.3/10a then m else -.</p> <p>c24: if D.4.3/10a and D.4.1/1b then x else -.</p> <p>c25: if D.4.3/10a then m else -.</p> <p>c26: if D.4.3/11a then m else -.</p> <p>c27: if D.4.3/11a and D.4.1/1b then x else -.</p> <p>c28: if D.4.3/12a then m else -.</p> <p>c29: if D.4.3/12a and D.4.1/1b then x else -.</p> <p>c30: if D.4.3/13a and D.4.1/1b then x else -.</p> <p>c31: if D.4.3/13a then m else -.</p> <p>c32: if D.4.3/14a and D.4.1/1b then x else -.</p> <p>c33: if D.4.3/14a then m else -.</p> <p>c34: if D.4.3/16a then m else -.</p> <p>c35: if D.4.3/16a and D.4.1/1b then x else -.</p>																

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	Sup- port	Addi- tional infor- mation	Subindex	Action field name label	Constraints and values	Status	Sup- port	Addi- tional infor- mation
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

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Addi-tional informa-tion
					Con-firm-ed	Non-con-firm-ed							
1	attribut eValueC hange	{dmi-not1}	c1				1.1	AttributeValu eChangeInfo		Information Syntax SEQUENCE	c1		
							1.1.1	sourceIndicat or	{dmi-att 26}	ENUMERATED	c:o		
							1.1.2	attributeIdentifi erList	{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	attributeValu eChangeDefin ition	{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	oldAttributeV alue	-	ANY DEFINED BY attributeID	c:o		
							1.1.3.3	newAttribut eValue	-	ANY DEFINED BY attributeID	c:m		
							1.1.4	notificationId entifier	{dmi-att 16}	INTEGER	c:o		
							1.1.5	correlatedNoti fications	{dmi-att 12}	SET OF SEQUENCE	c:o		
							1.1.5.1	correlatedNoti fications	{dmi-att 12}	SET OF INTEGER	c:m		
							1.1.5.2	sourceObjectInst	-	CHOICE	c:o		
							1.1.5.2.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o.3		
							1.1.5.2.3	localDistingui shedName	-	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		
							1.1.6	additionalText	{dmi-att 7}	GraphicString	c:o		
							1.1.7	additionalInfor mation	{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	Notifi- cation type template label	Value of object identifier for notification type	Con- straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-con-firm-ed								
							1.1.7.2	significance	-	BOOLEAN	c:o			
							1.1.7.3	information	-	ANY DEFINED BY identifier	c:m			
2	autoBac kupRepo rt	{2 9 2 18 10 1}	c2				2.1	AutoBackupR eportInfo		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	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o.5			
							2.1.1.2.1.3	localDistingui shedName	-	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	autoRest oreRepo rt	{2 9 2 18 10 2}	c3				3.1	AutoRestoreR eportInfo		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	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o.7			
							3.1.1.1.3	localDistingui shedName	-	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

		Support											
Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Con-firm-ed	Non-con-firm-ed	Additional information	Subindex	Notification field name label	Value of object identifier of attrib-ute type associated with field	Constraints and values	Status	Sup-port	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	objectCr-eation	{dmi-not6}	c4				4.1	ObjectInfo		Information Syntax SEQUENCE	c4		
							4.1.1	sourceIndicat-or	{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	notificationId entifier	{dmi-att 16}	INTEGER	c:o		
							4.1.4	correlatedNoti-fications	{dmi-att 12}	SET OF SEQUENCE	c:o		
							4.1.4.1	correlatedNoti-fications	{dmi-att 12}	SET OF INTEGER	c:m		
							4.1.4.2	sourceObjectI-nst	-	CHOICE	c:o		
							4.1.4.2.1	distinguishe-dName	-	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	nonSpecificFo-rm	-	OCTET STRING	c:o.9		
							4.1.4.2.3	localDistingui-shedName	-	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	additionalInfor-mation	{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

Index	Notifi- cation type template label	Value of object identifier for notification type	Support		Con- firm- ed	Non- con- firm- ed	Addi- tional infor- mation	Subindex	Notification field name label	Value of object identifier of attrib- ute type associated with field	Constraints and values	Status	Sup- port	Additional infor- mation
			Con- straints and values	Status										
								4.1.6.2	significance	-	BOOLEAN	c:o		
								4.1.6.3	information	-	ANY DEFINED BY identifier	c:m		
5	objectDe- letion	{dmi-not7}	c5					5.1	ObjectInfo		Information Syntax SEQUENCE	c5		
								5.1.1	sourceIndicat- or	{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	notificationId entifier	{dmi-att- 16}	INTEGER	c:o		
								5.1.4	correlatedNoti- fications	{dmi-att- 12}	SET OF SEQUENCE	c:o		
								5.1.4.1	correlatedNoti- fications	{dmi-att- 12}	SET OF INTEGER	c:m		
								5.1.4.2	sourceObjectI- nst	-	CHOICE	c:o		
								5.1.4.2.1	distinguishe- dName	-	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	nonSpecificFo- rm	-	OCTET STRING	c:o. 11		
								5.1.4.2.3	localDistingui- shedName	-	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	additionalInfor- mation	{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

Index	Notifi-cation type template label	Support		Con-firm-ed	Non-con-firm-ed	Addi-tional information	Subindex	Notification field name label	Value of object identifier of attrib-ute type associated with field	Constraints and values	Status	Sup-port	Additional information	
		Con-straints and values	Sta-tus											
6	processi ngErrorAlarm	{dmi-not10}	m	alarmEffectOnServiceParameter	6.1	AlarmInfo			Information Syntax SEQUENCE	m				
								6.1.1	probableCau se	{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	specificProble ms	{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	perceivedSeve rity	{dmi-att 17}	ENUMERATED	m		
								6.1.4	backedUpSta tus	{dmi-att 11}	BOOLEAN	o		
								6.1.5	backUpObject	{dmi-att 40}	CHOICE	o		
								6.1.5.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 14		
								6.1.5.3	localDistingui shedName	-	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	trendIndica tion	{dmi-att 30}	ENUMERATED	o		
								6.1.7	thresholdInfo	{dmi-att 29}	SEQUENCE	o		
								6.1.7.1	triggeredThre shold	-	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	observedVal ue	-	CHOICE	c:m		
								6.1.7.2.1	integer	-	INTEGER	c:o. 16		

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

Index	Notifi-cation type template label	Support				Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Sup-port	Additional information
		Value of object identifier for notification type	Con-strainsts and values	Sta-tus	Con-confirmed	Non-con-confirmed							
							6.1.7.2.2	real	-	REAL	c:o. 16		
							6.1.7.3	thresholdLev el	-	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	notificationId entifier	{dmi-att 16}	INTEGER	o		
							6.1.9	correlatedNoti fication	{dmi-att 12}	SET OF SEQUENCE	o		
							6.1.9.1	correlatedNoti fication	{dmi-att 12}	SET OF INTEGER	c:m		
							6.1.9.2	sourceObjectIn st	-	CHOICE	c:o		
							6.1.9.2.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 22		
							6.1.9.2.3	localDistingui shedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 22		

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

Index	Notifi-cation type template label	Value of object identifier for notification type	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attrib-ute type associated with field	Constraints and values	Status	Sup-port	Additional information
			Con-straints and values	Sta-tus								
						6.1.9.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
						6.1.9.2.3.2	AttributeValue	-	ANY	c:m		
						6.1.10	stateChan geDefinition	{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	oldAttribu teValue	-	ANY DEFINED BY attributeID	c:o		
						6.1.10.3	newAttribu teValue	-	ANY DEFINED BY attributeID	c:m		
						6.1.11	monitoredAttr ibutes	{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	proposedRepa irActions	{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	additionalInfo rmation	{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	stateCha nge	{dmi-not14}	c6			7.1	StateChan geInfo		Information Syntax SEQUENCE	c6		
						7.1.1	sourceIndicat or	{dmi-att 26}	ENUMERATED	c:o		
						7.1.2	attributeIdentifi erList	{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	Notifi- cation type template label	Value of object identifier for notification type	Con- straints and values	Sta-tus	Con-firm-ed	Non-con-firm-ed	Addi-tional information	Support	Subindex	Notification field name label	Value of object identifier of attrib-ute type associated with field				
												Constrains and values	Sta-tus	Sup- port	Additional information
									7.1.3	stateChan geDefinition	{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	oldAttribu teValue	-	ANY DEFINED BY attributeID	c:o		
									7.1.3.3	newAttribu teValue	-	ANY DEFINED BY attributeID	c:m		
									7.1.4	notificationId entifier	{dmi-att 16}	INTEGER	c:o		
									7.1.5	correlatedNoti fications	{dmi-att 12}	SET OF SEQUENCE	c:o		
									7.1.5.1	correlatedNoti fications	{dmi-att 12}	SET OF INTEGER	c:m		
									7.1.5.2	sourceObje ctInst	-	CHOICE	c:o		
									7.1.5.2.1	distinguishe dName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 28		
									7.1.5.2.3	localDistingui shedName	-	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	additionalInfo rmation	{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**

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	
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	fileType	{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	futureAutoRestoreAllocated	{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	
				Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support
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	distributedSoftware	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	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							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	Status	Support	Additional information	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
							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		

Table D.12 (concluded) – 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
7	validate	{swmf-act 6}	softwareProcessingFailureParameter	c6			7.1	ValidateInfo	Information Syntax CHOICE	c6		
<p>c1: if D.5.3/13a then m else –.</p> <p>c2: if D.5.2/15a then m else –.</p> <p>c3: if D.5.2/14a then m else –.</p> <p>c4: if D.5.2/19a then m else –.</p> <p>c5: if D.5.2/24a then m else –.</p> <p>c6: if D.5.2/28a then m else –.</p> <p>c7: if D.5.2/32a then m else –.</p>												

D.5.6 Notifications**Table D.13 – Executable Software Notification support**

Index	Notifi- cation type template label	Value of object identifier for notifica- tion type	Con- straints and values	Sta- tus	Support		Addi- tional infor- mation							
					Con- firm- ed	Non- con- firm- ed	Additional infor- mation	Subindex	Notification field name label	Value of object identifier of attribute type asso- ciated with field	Constrains and values	Sta- tus	Sup- port	Additional infor- mation
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	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		

Table D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-confirm-ed								
							1.1.5.2.3.2	AttributeValue	-	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			
							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			

Table D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Addi-tional infor-mation
					Con-firm-ed	Non-con-firm-ed								
							3.1.1.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.7			
							3.1.1.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m			
							3.1.1.1.2	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}	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			

Table D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-confirm-ed								
							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}	c5				5.1	ObjectInfo		Information Syntax SEQUENCE	c5			
							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 D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firm-ed	Non-con-firm-ed							
6	process-ingErr orAlarm	{dmi-not10}	m	alarmEffectOnServiceParameter	6.1	AlarmInfo	6.1.4.2	sourceObjectInst	-	CHOICE	c:o		
								distinguish edName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11		
								AttributeType	-	OBJECT IDENTIFIER	c:m		
								AttributeValue	-	ANY	c:m		
								nonSpecificFo rm	-	OCTET STRING	c:o. 11		
								localDistingui shedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11		
								AttributeType	-	OBJECT IDENTIFIER	c:m		
								AttributeValue	-	ANY	c:m		
								additionalText	{dmi-att 7}	GraphicString	c:o		
								additionalInfo rmation	{dmi-att 6}	SET OF SEQUENCE	c:o		
								identifier	-	OBJECT IDENTIFIER	c:m		
								significance	-	BOOLEAN	c:o		
								information	-	ANY DEFINED BY identifier	c:m		
					6.1.1	probableCau se	6.1.1.1	AlarmInfo		Information Syntax SEQUENCE	m		
								globalValue	{dmi-att 18}	CHOICE	m		
								localValue	-	OBJECT IDENTIFIER	o.12		
								specificProble ms	{dmi-att 27}	SET OF CHOICE	o		
								OBJECT IDENTIFIER	-	OBJECT IDENTIFIER	c:o. 13		
								INTEGER	-	INTEGER	c:o. 13		
								perceivedSeve rity	{dmi-att 17}	ENUMERATED	m		

Table D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	Support		Additional information	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Sup-port	Addi-tional infor-mation
		Con-straints and values	Sta-tus								
						6.1.4	backedUpStat us	{dmi-att 11}	BOOLEAN	o	
						6.1.5	backUpObject	{dmi-att 40}	CHOICE	o	
						6.1.5.1	distinguish edName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 14	
						6.1.5.3	localDistingui shedName	-	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	trendIndicati on	{dmi-att 30}	ENUMERATED	o	
						6.1.7	thresholdInfo	{dmi-att 29}	SEQUENCE	o	
						6.1.7.1	triggeredThre shold	-	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	observedVal ue	-	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	thresholdLev el	-	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 D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	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
		Con-straints and values	Sta-tus								
					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	notificationId entifier	{dmi-att 16}	INTEGER	o		
					6.1.9	correlatedNoti fication	{dmi-att 12}	SET OF SEQUENCE	o		
					6.1.9.1	correlatedNoti fication	{dmi-att 12}	SET OF INTEGER	c:m		
					6.1.9.2	sourceObjectI nst	-	CHOICE	c:o		
					6.1.9.2.1	distinguish edName	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 22		
					6.1.9.2.3	localDistingui shedName	-	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	stateChangeD efinition	{dmi-att 28}	SET OF SEQUENCE	o		

Table D.13 (continued) – Executable Software Notification support

Index	Notifi-cation type template label	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
		Con-strain-tions and values	Sta-tus								
					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.13 (concluded) – Executable Software Notification support

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional information
					Con-firmed	Non-con-firmed							
							7.1.3	stateChang eDefinition	{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	oldAttribut eValue	-	ANY DEFINED BY attributeID	c:o		
							7.1.3.3	newAttribute Value	-	ANY DEFINED BY attributeID	c:m		
							7.1.4	notificationId entifier	{dmi-att 16}	INTEGER	c:o		
							7.1.5	correlatedNoti fication	{dmi-att 12}	SET OF SEQUENCE	c:o		
							7.1.5.1	correlatedNoti fication	{dmi-att 12}	SET OF INTEGER	c:m		
							7.1.5.2	sourceObjec tInst	-	CHOICE	c:o		
							7.1.5.2.1	distinguished Name	-	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	nonSpecificFo rm	-	OCTET STRING	c:o. 28		
							7.1.5.2.3	localDistingui shedName	-	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	additionalInfo rmation	{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.5.3/5a then m else -.
c2: if D.5.3/11a then m else -.
c3: if D.5.3/12a then m else -.
c4: if D.5.3/7a then m else -.
c5: if D.5.3/7a then m else -.
c6: if D.5.3/23a then m else -.

D.5.7 Parameters

Table D.14 – Executable Software 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.5.3/13a or D.5.3/15a or D.5.3/19a or D.5.3/24a or D.5.3/28a or D.5.3/32a then m else –.						

D.6 Software distributor managed object class

D.6.1 Statement of conformance to the managed object class

The supplier of the implementation shall state whether or not all mandatory features of the software distributor 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.15 – Software Distributor 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	softwareDistributor	{swmf-obj 3}		

If the answer to the actual class question in Table D.15 is “N”, the supplier of the implementation shall supply the actual class support details, in Table D.17.

Table D.16 – Software Distributor 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.6.2 Packages**Table D.17 – Software Distributor Package support**

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	allomorphicPackage	{dmi-pkg 17}	“if an object supports allomorphism”	c1		
2	packagesPackage	{dmi-pkg 16}	“any registered package, other than this package has been instantiated”.	c2		
3	softwareDistributorPackage	(not registered)	Mandatory	m		
4	topPackage	(not registered)	Mandatory	m		
c1: if D.6.1/1b then – else m. c2: if D.6.2/2a then m else –.						

D.6.3 Attributes**Table D.18 – Software Distributor 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	
1	administrativeState	{dmi-att 31}		m		m		m		-		-		c1		
2	allomorphs	{dmi-att 50}	SET OF CHOICE	c2		c3		-		-		-		-		
3	nameBinding	{dmi-att 63}	OBJECT IDENTIFIER	o		m		x		-		-		x		
4	objectClass	{dmi-att 65}	CHOICE	m		m		x		-		-		x		
5	operationalState	{dmi-att 35}	ENUMERATED	x		m		x		-		-		x		
6	packages	{dmi-att 66}	SET OF OBJECT IDENTIFIER	o		c1		x		x		x		x		
7	softwareDistributorId	{swmf-att 21}	NameType	o		m		x		-		-		x		
c1: if D.6.1/1b then x else –. c2: if D.6.2/1a then o else –. c3: if D.6.2/1a then m else –.																

D.6.4 Attribute groups

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

D.6.5 Actions**Table D.19 – Software Distributor 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	deliver	{swmf-act 1}	softwareProcessingFailureParameter	m			1.1	DeliverInfo	Information Syntax SEQUENCE	m		

D.6.6 Notifications

Table D.20 – Software Distributor Notification support

Index	Notification type template label	Value of object identifier for notification type	Support				Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
			Con-	Sta-	Con-	Non-							
			straints and values	tus	firm ed	non- firm ed	ditional information						
1	deliverResultNotification	{swmf-not3}	m	m	m	m	1.1	DeliverResultInfo		Information Syntax SEQUENCE	m		
								1.1.1	deliverId	-	CHOICE	o	
								1.1.1.1	globalValue	-	OBJECT IDENTIFIER	c:o.1	
								1.1.1.2	localValue	-	INTEGER	c:o.1	
								1.1.2	deliverResult	-	ENUMERATED	m	
								1.1.3	additionalInfo	-	SET OF SEQUENCE	o	
								1.1.3.1	identifier	-	OBJECT IDENTIFIER	c:m	
								1.1.3.2	significance	-	BOOLEAN	c:o	
								1.1.3.3	information	-	ANY DEFINED BY identifier	c:m	
2	objectCreation	{dmi-not6}	m	m	m	m	2.1	ObjectInfo		Information Syntax SEQUENCE	m		
								2.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o	
								2.1.2	attributeList	{dmi-att 9}	SET OF SEQUENCE	c:o	
								2.1.2.1	attributeId	-	CHOICE	c:m	
								2.1.2.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.8	
								2.1.2.1.2	localForm	-	INTEGER	c:o.8	
								2.1.2.2	attributeValue	-	ANY DEFINED BY attributeId	c:m	
								2.1.3	notificationIdentifier	{dmi-att 16}	INTEGER	c:o	
								2.1.4	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o	
								2.1.4.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m	
								2.1.4.2	sourceObjectInst	-	CHOICE	c:o	
								2.1.4.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.9	
								2.1.4.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m	
								2.1.4.2.1.2	AttributeValue	-	ANY	c:m	
								2.1.4.2.2	nonSpecificForm	-	OCTET STRING	c:o.9	
								2.1.4.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o.9	
								2.1.4.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m	
								2.1.4.2.3.2	AttributeValue	-	ANY	c:m	

Table D.20 (continued) – Software Distributor Notification support

Index	Notifi-cation type template label	Value of object identifier for notification type	Con-straints and values	Sta-tus	Support		Addi-tional information	Subindex	Notification field name label	Value of object identifier of attrib-ute type associated with field	Constraints and values	Sta-tus	Sup-port	Additional infor-mation
					Con-firm-ed	Non-con-firm-ed								
							2.1.5	additionalText	{dmi-att 7}	GraphicString	c:o			
							2.1.6	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o			
							2.1.6.1	identifier	-	OBJECT IDENTIFIER	c:m			
							2.1.6.2	significance	-	BOOLEAN	c:o			
							2.1.6.3	information	-	ANY DEFINED BY identifier	c:m			
3	objectDeletion	{dmi-not7}	m				3.1	ObjectInfo		Information Syntax SEQUENCE	m			
							3.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o			
							3.1.2	attributeList	{dmi-att 9}	SET OF SEQUENCE	c:o			
							3.1.2.1	attributeId	-	CHOICE	c:m			
							3.1.2.1.1	globalForm	-	OBJECT IDENTIFIER	c:o. 10			
							3.1.2.1.2	localForm	-	INTEGER	c:o. 10			
							3.1.2.2	attributeValue	-	ANY DEFINED BY attributeId	c:m			
							3.1.3	notificationIdentifier	{dmi-att 16}	INTEGER	c:o			
							3.1.4	correlatedNotifications	{dmi-att 12}	SET OF SEQUENCE	c:o			
							3.1.4.1	correlatedNotifications	{dmi-att 12}	SET OF INTEGER	c:m			
							3.1.4.2	sourceObjectInst	-	CHOICE	c:o			
							3.1.4.2.1	distinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11			
							3.1.4.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m			
							3.1.4.2.1.2	AttributeValue	-	ANY	c:m			
							3.1.4.2.2	nonSpecificForm	-	OCTET STRING	c:o. 11			
							3.1.4.2.3	localDistinguishedName	-	SEQUENCE OF SET OF SEQUENCE	c:o. 11			
							3.1.4.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m			
							3.1.4.2.3.2	AttributeValue	-	ANY	c:m			
							3.1.5	additionalText	{dmi-att 7}	GraphicString	c:o			
							3.1.6	additionalInformation	{dmi-att 6}	SET OF SEQUENCE	c:o			
							3.1.6.1	identifier	-	OBJECT IDENTIFIER	c:m			
							3.1.6.2	significance	-	BOOLEAN	c:o			
							3.1.6.3	information	-	ANY DEFINED BY identifier	c:m			

Table D.20 (concluded) – Software Distributor Notification support

Index	Notifi- cation type template label	Support		Con- constraints and values	Sta- tus	Con- firm ed	Non- confirm ed	Addi- tional infor- mation	Subindex	Notification field name label	Value of object identifier of attrib- ute type associated with field	Constraints and values	Sta- tus	Sup- port	Addi- tional infor- mation	
4	stateChange	{dmi-not14}	m						4.1	StateChangeInfo		Information Syntax SE-QUENCE	m			
										4.1.1	sourceIndicator	{dmi-att 26}	ENUMERATED	c:o		
										4.1.2	attributeIdentifie rList	{dmi-att 8}	SET OF CHOICE	c:o		
										4.1.2.1	globalForm	-	OBJECT IDENTIFIER	c:o.	26	
										4.1.2.2	localForm	-	INTEGER	c:o.	26	
										4.1.3	stateChangeDefi nition	{dmi-att 28}	SET OF SEQUENCE	c:m		
										4.1.3.1	attributeID	-	CHOICE	c:m		
										4.1.3.1.1	globalForm	-	OBJECT IDENTIFIER	c:o.	27	
										4.1.3.1.2	localForm	-	INTEGER	c:o.	27	
										4.1.3.2	oldAttributeVal ue	-	ANY DEFINED BY attributeID	c:o		
										4.1.3.3	newAttributeVal ue	-	ANY DEFINED BY attributeID	c:m		
										4.1.4	notificationIdent ifier	{dmi-att 16}	INTEGER	c:o		
										4.1.5	correlatedNotifi cations	{dmi-att 12}	SET OF SEQUENCE	c:o		
										4.1.5.1	correlatedNotifi cations	{dmi-att 12}	SET OF INTEGER	c:m		
										4.1.5.2	sourceObjectInst	-	CHOICE	c:o		
										4.1.5.2.1	distinguishedNa me	-	SEQUENCE OF SET OF SEQUENCE	c:o.	28	
										4.1.5.2.1.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
										4.1.5.2.1.2	AttributeValue	-	ANY	c:m		
										4.1.5.2.2	nonSpecificFo rm	-	OCTET STRING	c:o.	28	
										4.1.5.2.3	localDistinguish edName	-	SEQUENCE OF SET OF SEQUENCE	c:o.	28	
										4.1.5.2.3.1	AttributeType	-	OBJECT IDENTIFIER	c:m		
										4.1.5.2.3.2	AttributeValue	-	ANY	c:m		
										4.1.6	additionalText	{dmi-att 7}	GraphicString	c:o		
										4.1.7	additionalInform ation	{dmi-att 6}	SET OF SEQUENCE	c:o		
										4.1.7.1	identifier	-	OBJECT IDENTIFIER	c:m		
										4.1.7.2	significance	-	BOOLEAN	c:o		
										4.1.7.3	information	-	ANY DEFINED BY identifier	c:m		

D.6.7 Parameters

There are no parameters specified for this managed object class.

Annexe E⁶⁾**Formulaire MRCS pour la corrélation de noms**

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

E.1 Introduction

The purpose of this MRCS proforma for name bindings is to provide a mechanism for a supplier which claims conformance, in the agent role, to a name binding to provide conformance information in a standard form.

E.2 Instructions for completing the MRCS proforma for name binding to produce an MRCS

The MRCS 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.

Notations used in the Status and Support columns are specified in A.1.3.

E.3 Symbols, abbreviations and terms

The following abbreviation is used in this MRCS proforma:

dmi-nb joint-iso-ccitt ms(9) smi(3) part2(2) nameBinding(6)

E.4 Statement of conformance to the name binding**Table E.1 – Name Binding support**

Index	Name binding template label	Value of object identifier for name binding	Constraints and values	Status	Support	Additional information
1	softwareUnit-subsystem	{2 9 2 18 6 2}	Superior class: “Rec. X.723 ISO/IEC 10165-5:1994”: subsystem AND SUBCLASSES	o		
2	softwareUnit-system	{2 9 2 18 6 4}	Superior class: “Rec. X.721 ISO/IEC 10165-2:1992”: system AND SUBCLASSES	o		
3	software-equipment	{0 0 13 3100 0 6 18}	Superior class: “Rec. M.3100:1995”: equipment AND SUBCLASSES	o		
4	software-managedElement	{0 0 13 3100 0 6 20}	Superior class: “Rec. M.3100:1995”: managedElement AND SUBCLASSES	o		
5	software-software	{0 0 13 3100 0 6 19}	Superior class: “Rec. M.3100:1995”: software AND SUBCLASSES	o		

⁶⁾ **Droits de reproduction du formulaire MRCS**

Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MRCS de la présente annexe pour utiliser celui-ci conformément à son objet. Ils sont également autorisés à publier le formulaire une fois celui-ci complété. Les instructions pour compléter le formulaire MRCS sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 10165-6.

Table E.1 (concluded)– Name Binding support

Subindex	Operation	Constraints and values	Status	Support	Additional information
1.1	Create support		c:m		
1.2	Create with reference object		-		
1.3	Create with automatic instance naming		c:m		
1.4	Delete support		c:m		
1.5	Delete only if no contained objects		c:x		
1.6	Delete contained objects		c:m		
2.1	Create support		c:m		
2.2	Create with reference object		-		
2.3	Create with automatic instance naming		c:m		
2.4	Delete support		c:m		
2.5	Delete only if no contained objects		c:x		
2.6	Delete contained objects		c:m		
3.1	Create support		c:m		
3.2	Create with reference object		c:m		
3.3	Create with automatic instance naming		c:m		
3.4	Delete support		c:m		
3.5	Delete only if no contained objects		c:m		
3.6	Delete contained objects		c:x		
4.1	Create support		c:m		
4.2	Create with reference object		c:m		
4.3	Create with automatic instance naming		c:m		
4.4	Delete support		c:m		
4.5	Delete only if no contained objects		c:m		
4.6	Delete contained objects		c:x		
5.1	Create support		c:m		
5.2	Create with reference object		c:m		
5.3	Create with automatic instance naming		c:m		
5.4	Delete support		c:m		
5.5	Delete only if no contained objects		c:m		
5.6	Delete contained objects		c:x		

Annexe F⁷⁾**Formulaire MIDS**

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

F.1 Actions**Table F.1 – 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	o			1.1	BackupArgument	Information Syntax SEQUENCE	c:m		
							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	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		
2	restore	{swmf-act 8}		o			2.1	RestoreArgument	Information Syntax SEQUENCE	c:m		
							2.1.1	restoreSource	CHOICE	c:m		
							2.1.1.1	localObject	Object Instance	c:o.1		
							2.1.1.2	inLine	BIT STRING	c:o.1		
							2.1.1.3	offLine	Printable String	c:o.1		
							2.1.2	additionalInfo	SET OF ANY	c:o		

⁷⁾ Droits de reproduction du formulaire MIDS

Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MIDS de la présente annexe pour utiliser celui-ci conformément à son objet. Ils sont également autorisés à publier le formulaire une fois celui-ci complété. Les instructions pour compléter le formulaire MIDS sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 10165-6.

Annexe G**Classe d'objets logiciels de la Recommandation M.3100**

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

NOTE – This annex repeats the GDMO templates associated with the Software object class specified in Recommendation M.3100. The templates from Recommendation M.3100 are included in this Specification for information only. Templates associated with the Software object class which are specified in CCITT Rec. X.721 | ISO/IEC 10165-2 are not included here.

G.1 Software Object Class**software MANAGED OBJECT CLASS**

```

DERIVED FROM "Recommendation X.721:1992":top;
CHARACTERIZED BY softwarePackage PACKAGE
BEHAVIOUR softwareBehaviour ;
ATTRIBUTES
softwareId
GET;;
CONDITIONAL PACKAGES
createDeleteNotificationsPackage
PRESENT IF "the objectCreation and objectDeletion notifications defined
in Recommendation X.721 are supported by an instance of this class.",

attributeValueChangeNotificationPackage
PRESENT IF "the attributeValueChange notification defined in
Recommendation X.721 is supported by an instance of this class.",

stateChangeNotificationPackage
PRESENT IF "the stateChange notification defined in
Recommendation X.721 is supported by an instance of this class.",

administrativeOperationalStatesPackage
PRESENT IF "an instance supports it.",

affectedObjectListPackage
PRESENT IF "an instance supports it.",

softwareProcessingErrorAlarmPackage
PRESENT IF "an instance supports it.",

userLabelPackage
PRESENT IF "an instance supports it.",

vendorNamePackage
PRESENT IF "an instance supports it.",

versionPackage
PRESENT IF "an instance supports it.",

currentProblemListPackage
PRESENT IF "an instance supports it";

REGISTERED AS {m3100ObjectClass 4};

```

softwareBehaviour BEHAVIOUR**DEFINED AS**

"The Software object class is a class of managed objects that represent logical information stored in equipment, including programs and data tables. Software may be nested within other software, thereby creating a containment relationship. When the attribute value change notification package is present, the attributeValueChange notification defined in Recommendation X.721 shall be emitted when the value of one of the following attribute changes: alarm status, affected object list, user label, version, and current problem list. Because the above attributes are all in conditional packages, the behaviour for emitting the attribute value change notification applies only when the corresponding conditional packages are present in the managed object. When the state change notification package is present, the stateChangeNotification defined in Recommendation X.721 shall be emitted if the value of administrative state or operational state changes (when the administrativeOperationalStates conditional package is present).";

G.2 Packages**G.2.1 Administrative Operational States Package****administrativeOperationalStatesPackage PACKAGE**

```

ATTRIBUTES
"Recommendation X.721:1992":administrativeState
GET-REPLACE,
"Recommendation X.721:1992":operationalState
GET;
REGISTERED AS {m3100Package 1};

```

G.2.2 Affected Object List Package

```
affectedObjectListPackage PACKAGE
  ATTRIBUTES
    affectedObjectList
      GET;
  REGISTERED AS {m3100Package 2};
```

G.2.3 Attribute Value Change Notification Package

```
attributeValueChangeNotificationPackage PACKAGE
  NOTIFICATIONS
    "Recommendation X.721:1992":attributeValueChange;
  REGISTERED AS {m3100Package 4};
```

G.2.4 Create Delete Notifications Package

```
createDeleteNotificationsPackage PACKAGE
  NOTIFICATIONS
    "Recommendation X.721:1992":objectCreation,
    "Recommendation X.721:1992":objectDeletion;
  REGISTERED AS {m3100Package 10};
```

G.2.5 Current Problem List Package

```
currentProblemListPackage PACKAGE
  ATTRIBUTES
    currentProblemList
      GET;
  REGISTERED AS {m3100Package 13};
softwareProcessingErrorAlarmPackage PACKAGE
  ATTRIBUTES
    alarmStatus
      GET;
  NOTIFICATIONS
    "Recommendation X.721:1992":processingErrorAlarm;
  REGISTERED AS {m3100Package 26};
```

G.2.6 Software Processing Error Alarm Package

```
softwareProcessingErrorAlarmPackage PACKAGE
  ATTRIBUTES
    alarmStatus
      GET;
  NOTIFICATIONS
    "Recommendation X.721:1992":processingErrorAlarm;
  REGISTERED AS {m3100Package 26};
```

G.2.7 State Change Notification Package

```
stateChangeNotificationPackage PACKAGE
  NOTIFICATIONS
    "Recommendation X.721:1992":stateChange;
  REGISTERED AS {m3100Package 28};
```

G.2.8 User Label Package

```
userLabelPackage PACKAGE
  ATTRIBUTES
    userLabel
      GET-REPLACE;
  REGISTERED AS {m3100Package 32};
```

G.2.9 Vendor Name Package

```
vendorNamePackage PACKAGE
  ATTRIBUTES
    vendorName
      GET-REPLACE;
  REGISTERED AS {m3100Package 33};
```

G.2.10 Version Package

```

versionPackage PACKAGE
  ATTRIBUTES
    version
      GET-REPLACE;
REGISTERED AS {m3100Package 34};

```

G.3 Attributes**G.3.1 Affected Object List**

```

affectedObjectList ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectList;
  MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
  BEHAVIOUR affectedObjectListBehaviour ;
REGISTERED AS {m3100Attribute 2};

```

-- *ObjectList ::= SET OF ObjectInstance*

```

affectedObjectListBehaviour
  BEHAVIOUR
    DEFINED AS

```

"The Affected Object List attribute type specifies the object instances which can be directly affected by a change in state or deletion of a given managed object. The attribute does not force internal details to be specified, but only the necessary level of detail required for management.";

```

  affectedObjectListPackage PACKAGE
    ATTRIBUTES
      affectedObjectList
        GET;
REGISTERED AS {m3100Package 2};

```

G.3.2 Alarm Status

```

alarmStatus ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.AlarmStatus;
  MATCHES FOR EQUALITY;
  BEHAVIOUR alarmStatusBehaviour;
REGISTERED AS {m3100Attribute 6};

```

-- *AlarmStatus ::= SET OF INTEGER { underRepair(0), critical(1), major(2), minor(3), alarmOutstanding(4) }*

```

alarmStatusBehaviour BEHAVIOUR
  DEFINED AS

```

"The Alarm Status attribute type indicates the occurrence of an abnormal condition relating to an object. This attribute may also function as a summary indicator of alarm conditions associated with a specific resource. It is used to indicate the existence of an alarm condition, a pending alarm condition such as threshold situations, or (when used as a summary indicator) the highest severity of active alarm conditions. When used as a summary indicator, the order of severity (from highest to lowest) is: activeReportable-Critical activeReportable-Major activeReportable-Minor activeReportable-Indeterminate activeReportable-Warning activePending cleared";

```

  attributeValueChangeNotificationPackage PACKAGE
    NOTIFICATIONS
      "Recommendation X.721:1992":attributeValueChange;
REGISTERED AS {m3100Package 4};

```

G.3.3 Current Problem List

```

currentProblemList ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.CurrentProblemList;
  BEHAVIOUR currentProblemListBehaviour;
REGISTERED AS {m3100Attribute 17};

```

```

-- CurrentProblem ::= SEQUENCE {
--   problem [0] ProbableCause,
--   alarmStatus [1] AlarmStatus}
-- CurrentProblemList ::= SET OF CurrentProblem

```

```
-- AlarmStatus ::= ENUMERATED {
--   cleared(0),
--   activeReportable-Indeterminate(1),
--   activeReportable-Warning(2),
--   activeReportable-Minor(3),
--   activeReportable-Major(4),
--   activeReportable-Critical(5),
--   activePending(6)}
-- ProblemCause ::= CHOICE {
--   unknown NULL,
--   integerValue INTEGER}

-- The values of integerValue for ProblemCause and integerValue for ResourceProblem shall always be
-- assigned by this Recommendation. No values of integerValue for ResourceProblem have been assigned.
-- The following values are used for integerValue of ProblemCause.

-- noSuchTpInstance ProblemCause ::= integerValue : 0
-- noSuchGtpInstance ProblemCause ::= integerValue : 1
-- noSuchTpPoolInstance ProblemCause ::= integerValue : 2
-- mismatchingTpInstance ProblemCause ::= integerValue : 3
-- mismatchingGtpInstance ProblemCause ::= integerValue : 4
-- partOfGtp ProblemCause ::= integerValue : 5
-- involvedInCrossConnection ProblemCause ::= integerValue : 6
-- memberOfTpPool ProblemCause ::= integerValue : 7
-- alreadyMemberOfGtp ProblemCause ::= integerValue : 8
-- noTpInTpPool ProblemCause ::= integerValue : 9
-- noMoreThanOneTpIsAllowed ProblemCause ::= integerValue : 10
-- noMoreThanTwoTpsAreAllowed ProblemCause ::= integerValue : 11
```

currentProblemListBehaviour**BEHAVIOUR****DEFINED AS**

"The Current Problem List attribute type identifies the current existing problems, with severity, associated with the managed object.";

currentProblemListPackage PACKAGE

ATTRIBUTES**currentProblemList**

GET;

REGISTERED AS {m3100Package 13};

G.3.4 Software Id**softwareId ATTRIBUTE**

WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;

MATCHES FOR EQUALITY;

BEHAVIOUR softwareIdBehaviour;

REGISTERED AS {m3100Attribute 38};

```
-- NameType ::= CHOICE {
--   numericName INTEGER,
--   pString GraphicString}
```

softwareIdBehaviour BEHAVIOUR**DEFINED AS**

"The Software Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Software object class.";

softwareProcessingErrorAlarmPackage PACKAGE

ATTRIBUTES**alarmStatus**

GET;

NOTIFICATIONS

"Recommendation X.721:1992":processingErrorAlarm;

REGISTERED AS {m3100Package 26};

G.3.5 User Label

```
userLabel ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.UserLabel;
  MATCHES FOR EQUALITY, SUBSTRINGS;
  BEHAVIOUR userLabelBehaviour;
REGISTERED AS {m3100Attribute 50};

-- UserLabel ::= GraphicString

userLabelBehaviour BEHAVIOUR
  DEFINED AS
  "The User Label attribute type assigns a user friendly name to the associated object.";
  userLabelPackage PACKAGE
  ATTRIBUTES
    userLabel
      GET-REPLACE;
REGISTERED AS {m3100Package 32};
```

G.3.6 Vendor Name

```
vendorName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.VendorName;
  MATCHES FOR EQUALITY, SUBSTRINGS;
  BEHAVIOUR vendorNameBehaviour;
REGISTERED AS {m3100Attribute 51};

-- VendorName ::= GraphicString

vendorNameBehaviour BEHAVIOUR
  DEFINED AS
  "The Vendor Name attribute type identifies the vendor of the associated managed object.";
  vendorNamePackage PACKAGE
  ATTRIBUTES
    vendorName
      GET-REPLACE;
REGISTERED AS {m3100Package 33};
```

G.3.7 Version

```
version ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.Version;
  MATCHES FOR EQUALITY, SUBSTRINGS;
  BEHAVIOUR versionBehaviour;
REGISTERED AS {m3100Attribute 52};

-- Version ::= GraphicString

versionBehaviour BEHAVIOUR
  DEFINED AS
  "The Version attribute type identifies the version of the associated managed object.";
  versionPackage PACKAGE
  ATTRIBUTES
    version
      GET-REPLACE;
REGISTERED AS {m3100Package 34};
```

Annexe H

Mise en correspondance du modèle POSIX et du modèle de gestion de logiciel

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Le modèle POSIX est constitué des objets suivants:

- **Distribution (*distribution*):** une distribution se compose d'un ou de plusieurs produits se trouvant sous une forme prête à l'installation. La distribution de logiciel existe au niveau le plus élevé. Une distribution peut être décrite comme une collection de produits pouvant se trouver sur un media de distribution ou faire partie du stockage de fichiers d'un système. La distribution peut ne pas être soumise aux contraintes de produits ou d'architecture.
- **Installed Software (*logiciel installé*):** un logiciel installé se compose d'un ou de plusieurs produits se présentant sous une forme prête à l'utilisation. Pour obtenir un objet logiciel installé, l'administrateur installe un ou plusieurs produits pris dans une distribution puis les configure. Le logiciel installé a un niveau équivalent à celui d'une distribution mais après que les produits logiciels aient été installés sur un système de fichier. A noter que le processus de configuration sort du cadre du document POSIX.
- **Product (*produit*):** le niveau produit est le niveau d'agrégation le plus élevé pour des logiciels qui sont en relation. Le niveau produit définit le ou les systèmes d'exploitation et la ou les architectures matérielles prises en charge par le produit. Ces attributs s'appliquent à l'ensemble des sous-produits et des ensembles de fichiers contenus dans un produit. Un produit contient des ensembles de fichiers qui peuvent être regroupés d'une manière optionnelle en sous-produits.
- **Fileset (*ensemble de fichiers*):** un ensemble de fichiers est un groupe de fichiers. Le niveau situé au-dessus des produits est celui décrivant les ensembles de fichiers. L'ensemble de fichiers est l'unité de logiciel chargée et gérée par le fournisseur de produit logiciel. C'est également la plus petite unité que l'administrateur de système peut choisir de charger. Les ensembles de fichiers contiennent les fichiers constituant le produit. Ils contiennent également les descriptions de fichiers et les scénarios adaptés utilisés par les outils d'installation, de mise à jour et de suppression permettant d'installer et de désinstaller le logiciel.
- **Subproduct (*sous-produit*):** les sous-produits sont uniquement une notation commode pour une agrégation d'ensembles de fichiers. Des sous-produits peuvent contenir d'autres sous-produits ou des ensembles de fichiers ou une combinaison des deux, à la seule condition de pouvoir être décomposés en ensembles de fichiers. Un sous-produit n'a pas de comportement propre et constitue uniquement une façon commode de spécifier des ensembles de fichiers. Le niveau de sous-produit est optionnel. Les sous-produits fournissent un niveau intermédiaire d'organisation pour des ensembles de fichiers contenant de nombreux ensembles de fichiers. Des niveaux multiples de sous-produits sont pris en charge, un sous-produit peut contenir d'autres sous-produits et un ensemble de fichiers peut appartenir à plus d'un sous-produit.
- **Control Script (*script de commande*):** un script de commande est un fichier exécutable. Les scripts de commande fournissent des fonctions de vérification avant et après installation, etc. Les scénarios de commande sont les fichiers de description de produit décrivant des ensembles de fichiers INDEX et INFO ainsi que des scripts de commande fournis par le vendeur. Les fichiers INDEX décrivent la structure du produit pour un ensemble de fichiers. Les fichiers INFO décrivent les fichiers correspondants et leurs attributs. Les scripts de commande contiennent les opérations de préparation, de terminaison, de fin et de personnalisation pour l'installation et la désinstallation.
- **File (*fichier*):** les fichiers constituent le niveau le plus bas pour les objets faisant partie d'une distribution ou d'une installation. En plus des fichiers constituant l'ensemble de fichiers, sont présents des scripts de commande permettant d'installer et d'enlever des produits et des ensembles de fichiers. Les fichiers sont les fichiers effectifs et les répertoires des ensembles de fichiers contenant des attributs de fichiers tels que le propriétaire, le groupe et les permissions.

- **Bundle (lot):** un lot est un objet logiciel utilisé pour construire des agrégats d'autres objets logiciels contenant tout ou partie d'autres lots ou produits. Un lot est un groupement d'objets logiciels. Les lots ne contiennent pas d'objets logiciels appartenant à l'espace de noms du lot mais font référence à des objets logiciels en utilisant des attributs. Des lots peuvent faire référence à d'autres lots. Un objet logiciel peut être référencé par plus d'un lot. Le tableau suivant donne la correspondance éventuelle entre les commandes logicielles de POSIX et les commandes comparables de la présente Recommandation | Norme internationale.

Tableau H.1 – Correspondance entre les commandes logicielles POSIX et les opérations sur les unités logicielles

Commandes POSIX	Opérations ISO
swask	en dehors du domaine
swconfig	INSTALL/utilize
swcopy	CREATE ou DELIVER (Note)
swinstall	INSTALL
swlist	GET
swmodify	SET
swpackage	en dehors du domaine
swremove	DELETE
swverify	VALIDATE

NOTE – La présente Recommandation | Norme internationale utilise la distribution de logiciel et l'unité de logiciel pour la livraison de logiciel, alors que POSIX n'utilise que la distribution de logiciel. Pour POSIX, une distribution est un objet statique contenant un ensemble d'objets logiciels. Pour la présente Recommandation | Norme internationale, une distribution est un objet plus dynamique qui gère la distribution d'unités de logiciel d'un système vers un autre. Ce concept est défini dans POSIX comme une tâche répartie «copy» ou «install». Le concept de tâche POSIX signifie généralement l'application d'un ensemble d'objets logiciels à un ensemble d'objets système cibles

La distribution de logiciel POSIX correspond à une unité de logiciel ISO. L'unité de logiciel ISO correspond aux objets POSIX produit, ensemble de fichiers et fichiers.

Le Tableau H.2 donne la correspondance entre les objets logiciels POSIX et les classes d'objets définis par la présente Recommandation | Norme internationale. Il n'existe pas d'objet POSIX correspondant à la fonctionnalité de classe d'objets de distribution de logiciel de la présente Recommandation | Norme internationale.

Tableau H.2 – Correspondance entre les objets logiciels POSIX et les classes d'objets gérés logiciels

Classe d'objets POSIX	Classe d'objets ISO
distribution de logiciel	unité de logiciel (<i>software unit</i>)
produit	unité de logiciel
ensemble de fichiers	unité de logiciel
fichier	unité de logiciel
lot	nécessite une étude complémentaire

NOTE – La classe d'objets ISO correspondant à la classe d'objets POSIX «lot» appelle une étude complémentaire. (Elle pourrait correspondre à une unité logicielle.)

Le Tableau H.3 donne la correspondance éventuelle entre les attributs POSIX et les attributs comparables de l'unité de logiciel définis par la présente Recommandation | Norme internationale.

Tableau H.3 – Correspondance entre les attributs logiciels POSIX et les attributs d'unité logicielle

Attributs d'unité logicielle ISO	Attributs Fileset POSIX (sauf indication contraire)
softwareID	software_spec (un identificateur d'ensemble de fichiers universellement unique, incluant un préfixe de produit)
userLabel	title
vendorName	vendor_title (in product)
version	revision
currentProblem	–
operationalState	state
administrativeState	state
proceduralStatus	state
availabilityStatus	state
fileType	type (in file)
fileLocation	path (in file)
fileSize	size (in file)
checkSum	cksum (in file)
noteField	description
dateOfCreation	–
identityOfCreator	–
dateOfLastModification	mtime
identityOfLastModifier	–
dateDelivered	–
dateInstalled	create_time

SERIES DES RECOMMANDATIONS UIT-T

- | | |
|----------------|--|
| Série A | Organisation du travail de l'UIT-T |
| Série B | Moyens d'expression |
| Série C | Statistiques générales des télécommunications |
| Série D | Principes généraux de tarification |
| Série E | Réseau téléphonique et RNIS |
| Série F | Services de télécommunication non téléphoniques |
| Série G | Systèmes et supports de transmission |
| Série H | Transmission des signaux autres que téléphoniques |
| Série I | Réseau numérique à intégration de services |
| Série J | Transmission des signaux radiophoniques et télévisuels |
| Série K | Protection contre les perturbations |
| Série L | Construction, installation et protection des câbles et autres éléments des installations extérieures |
| Série M | Maintenance: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux |
| Série N | Maintenance: circuits internationaux de transmission radiophoniques et télévisuels |
| Série O | Spécifications des appareils de mesure |
| Série P | Qualité de transmission téléphonique |
| Série Q | Commutation et signalisation |
| Série R | Transmission télégraphique |
| Série S | Equipements terminaux de télégraphie alphabétique |
| Série T | Equipements terminaux et protocoles des services télématiques |
| Série U | Commutation télégraphique |
| Série V | Communications de données sur le réseau téléphonique |
| Série X | Réseaux pour données et communication entre systèmes ouverts |
| Série Z | Langages de programmation |