



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**Q.834.4**

**Amendement 1**  
(01/2004)

SÉRIE Q: COMMUTATION ET SIGNALISATION  
Interface Q3

---

Spécification d'une interface en architecture  
CORBA pour les réseaux optiques passifs à large  
bande, basée sur les prescriptions d'interface UML  
**Amendement 1**

Recommandation UIT-T Q.834.4 (2003) – Amendement 1

---

RECOMMANDATIONS UIT-T DE LA SÉRIE Q  
COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4, 5, 6, R1 ET R2	Q.120–Q.499
COMMULATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.799
<b>INTERFACE Q3</b>	<b>Q.800–Q.849</b>
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1699
PRESCRIPTIONS ET PROTOCOLES DE SIGNALISATION POUR LES IMT-2000	Q.1700–Q.1799
SPÉCIFICATIONS DE LA SIGNALISATION RELATIVE À LA COMMANDE D'APPEL INDÉPENDANTE DU SUPPORT	Q.1900–Q.1999
RNIS À LARGE BANDE	Q.2000–Q.2999

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

## Recommandation UIT-T Q.834.4

### Spécification d'une interface en architecture CORBA pour les réseaux optiques passifs à large bande, basée sur les prescriptions d'interface UML

#### Amendement 1

##### Résumé

Le présent amendement propose plusieurs modifications de la spécification de l'interface CORBA pour l'interface qui, entre un système de gestion par fournisseur et un système de gestion par opérateur, gère des réseaux optiques passifs à large bande (BPON, *broadband passive optical network*). Il englobe les modifications nécessaires pour la gestion de l'attribution dynamique de largeur de bande et pour la télémétrie automatique des unités de réseau optique (ONU, *optical network unit*) et des terminaisons de réseau optique (ONT, *optical network terminal*). Enfin il propose les adjonctions nécessaires pour la prise en charge adéquate des tests des services téléphoniques et pour données, la notification de l'activité du logiciel et la gestion de la capacité.

##### Source

L'Amendement 1 de la Recommandation Q.834.4 (2003) de l'UIT-T a été approuvé le 13 janvier 2004 par la Commission d'études 4 (2001-2004) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

## 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. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies 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.

## 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.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2004

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

## TABLE DES MATIÈRES

		Page
1	Domaine d'application .....	1
2	Références normatives.....	1
3	Termes et définitions .....	1
	3.1 Termes importés de la Rec. UIT-T G.983.4.....	1
	3.2 Termes nouveaux.....	2
4	Abréviations.....	2
5	Conventions.....	2
6	Améliorations des interfaces Build et Common.....	2
	6.1 Modifications du § 9.2.1.1, "buildNode".....	2
	6.2 Modifications du § 9.2.1.2, "assignUserLabelsToNE" .....	3
	6.3 Modifications du § 9.2.1.3, "modifyNode" .....	3
	6.4 buildTCONT.....	4
	6.5 modifyTCONTParameters .....	4
	6.6 deleteTCONT .....	5
	6.7 Modifications du § 9.2.1.20, "Exceptions".....	5
7	Améliorations de l'interface de test .....	5
	7.1 metallicDropTest .....	6
	7.2 scheduleMetallicDropTest.....	6
	7.3 mACLAYERTest .....	6
	7.4 scheduleMACLayerTest.....	7
	7.5 drawDialToneBreakTest.....	7
	7.6 scheduleDrawDialToneBreakTest.....	8
	7.7 Modifications du § 9.15.1.5, "modifyResourceSelfTestSchedule" .....	8
	7.8 Modifications du § 9.15.1.6, "cancelScheduledResourceSelfTest" .....	9
	7.9 Modifications du § 9.15.1.7, "conductResourceSelfTest".....	9
8	Améliorations de l'interface fournisseur d'événement.....	9
	8.1 ActivityCompletionEventSupplier .....	9
	8.2 Tableau 1, "q834_4 module organisation" .....	10
9	Améliorations de l'interface Q.834 Common pour la gestion des capacités .....	10
10	Améliorations de l'Annexe A "Dictionnaire de données" .....	11
11	Améliorations de l'Annexe B, "Exceptions".....	11
12	Améliorations de l'Annexe C, "Fichiers en langage IDL" .....	11
	12.1 Améliorations du § C.2, "Q834Build.idl" .....	11
	12.2 Améliorations du § C.3, "Q834Common.idl" .....	15
	12.3 Améliorations du § C.6, "Q834Eventpublisher.idl" .....	18

	<b>Page</b>
12.4 Améliorations du § C.15, "Q834Test.idl" .....	19
13 Améliorations de l'Annexe D, "Exemple de gabarits d'extrémité" .....	22

## Recommandation UIT-T Q.834.4

### Spécification d'une interface en architecture CORBA pour les réseaux optiques passifs à large bande, basée sur les prescriptions d'interface UML

#### Amendement 1

##### 1 Domaine d'application

Le présent amendement propose plusieurs modifications de la spécification de l'interface CORBA pour l'interface qui, entre un système de gestion par fournisseur et un système de gestion par opérateur, gère des réseaux optiques passifs à large bande (BPON). Il englobe les modifications nécessaires pour la gestion de l'attribution dynamique de largeur de bande et pour la télémétrie automatique des unités de réseau optique (ONU) et des terminaisons de réseau optique (ONT). Enfin, il propose les adjonctions nécessaires pour la prise en charge adéquate des tests des services téléphoniques et pour données, la notification de l'activité du logiciel et la gestion de la capacité.

##### 2 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

- [1] Recommandation UIT-T G.983.4 (2001), *Système d'accès optique à large bande avec capacité de service accrue par assignation dynamique de largeur de bande.*
- [2] Recommandation UIT-T G.983.7 (2001), *Spécification de l'interface de gestion et de commande de terminaison optique pour système de réseau optique passif à large bande avec attribution dynamique de largeur de bande.*
- [3] Recommandation UIT-T G.983.4 (2001)/Amd.1 (2003), *Système d'accès optique à large bande avec capacité de service accrue par assignation dynamique de largeur de bande – Nouvelle Annexe A – Paramètre de surveillance de la qualité de service.*
- [4] Recommandation UIT-T G.983.1 (1998), *Systèmes d'accès optique à large bande basés sur un réseau optique passif.*

##### 3 Termes et définitions

Pour les besoins de la présente Recommandation, les termes suivants s'appliquent:

###### 3.1 Termes importés de la Rec. UIT-T G.983.4

Les termes suivants repris de la Rec. UIT-T G.983.4 [1] sont utilisés dans le présent amendement:

- attribution dynamique de bande;
- T-CONT;
- largeur de bande garantie;

- largeur de bande fixe;
- largeur de bande assurée;
- largeur de bande maximale.

### 3.2 Termes nouveaux

Le présent amendement ne définit aucun terme nouveau.

## 4 Abréviations

Aucune abréviation autre que celles contenues dans la Recommandation de base n'est utilisée dans le présent amendement.

## 5 Conventions

Le présent amendement ne s'appuie sur aucune convention autre que celles de la Recommandation de base.

## 6 Améliorations des interfaces Build et Common

Le présent paragraphe porte sur les modifications des interfaces Build and Common de la Rec. UIT-Q.834 nécessaires pour prendre en charge la gestion de l'attribution dynamique de largeur de bande ainsi que la télémétrie automatique des terminaisons ONT et unités ONU conformément à l'un des mécanismes de télémétrie définis dans la Rec. UIT-T G.983.1. En premier lieu, les opérations utilisées pour construire ou pour modifier des attributs pour des éléments de réseau ONT et ONU sont modifiées afin d'inclure les paramètres d'entrée nécessaires pour caractériser correctement de tels éléments assurant l'attribution dynamique de largeur de bande et la télémétrie automatique. Ensuite, de nouvelles opérations sont ajoutées pour conférer la capacité de construction des TCONT, un partitionnement de la largeur de bande en vue de l'utilisation au niveau d'un élément ONT ou ONU en association avec les mécanismes de protocole d'attribution dynamique de largeur de bande. Ces opérations sont appelées buildTCONT, modifyTCONTParameters et deleteTCONT. Les nouvelles opérations créent le besoin d'une nouvelle exception appelée UnsupportedTCONTType, l'amélioration de la définition de l'exception ParameterViolation et l'utilisation nouvelle d'une exception existante, InsufficientPONBW, par l'interface Build.

Enfin les structures de données utilisées pour marquer les éléments ONT et ONU dans l'interface Common de Q.834 sont modifiées pour englober ces nouveaux attributs et de nouveaux types d'enregistrement HistoryData sont ajoutés l'interface Common pour permettre le contrôle des performances de la fonction d'attribution dynamique de largeur de bande. Ils sont appelés DBAFairnessPMHistoryDataType et TCONTTEPMHistoryDataType.

Toutes les modifications de l'interface Common et les modifications associées de l'Annexe A et de l'Annexe B sont combinées avec d'autres améliorations et présentées respectivement aux § 12.2, 10 et 11 du présent amendement.

### 6.1 Modifications du § 9.2.1.1, "buildNode"

a) *Remplacer:*

```
"
    in AdministrationDomainType administrationDomain)
    raises (UnrecognisedVersion, InvalidSerialNumSyntax,
    DuplicateSerialNumber, UnknownProfiles,"
```

*par:*

```
"          in RegistrationIdType registrationId,
          in boolean sRInd,
          in AdministrationDomainType administrationDomain)
          raises (UnrecognisedVersion, InvalidSerialNumSyntax,
          DuplicateSerialNumber, UnknownProfiles, ParameterViolation,"
```

b) *Remplacer:*

"L'entrée **administrationDomain** désigne le domaine auquel l'élément de réseau appartient."

*par:*

"L'entrée **registrationId** constitue la clé logique pour associer ces données de configuration à un élément de réseau installé. L'entrée **sRInd** spécifie si l'élément de réseau doit ou ne doit pas signaler son statut quand l'élément de réseau construit est une terminaison ONT ou une unité ONU et que l'attribution dynamique de largeur de bande est assurée. L'entrée **administrationDomain** désigne le domaine auquel appartient l'élément de réseau."

## 6.2 Modifications du § 9.2.1.2, "assignUserLabelsToNE"

a) *Remplacer:*

```
"          in AdministrationDomainType administrationDomain)"
```

*par:*

```
"          in RegistrationIdType registrationId,
          in AdministrationDomainType administrationDomain)"
```

b) *Remplacer:*

"L'entrée **administrationDomain** désigne le domaine auquel l'élément de réseau appartient."

*par:*

"L'entrée **registrationId** constitue la clé logique pour associer ces données de configuration à un élément de réseau installé. L'entrée **administrationDomain** désigne le domaine auquel appartient l'élément de réseau."

## 6.3 Modifications du § 9.2.1.3, "modifyNode"

a) *Remplacer:*

```
"          in AdministrationDomainType administrationDomain)
          raises (UnknownManagedEntity, UnknownNE,
          InvalidSlotAssignmentList, UnknownProfiles,
          DuplicateUserLabel, AccessDenied, InvalidExternalTime,
          ProfileSuspended);"
```

*par:*

```
"          in RegistrationIdType registrationId,
          in boolean sRInd,
          in AdministrationDomainType administrationDomain)
          raises (UnknownManagedEntity, UnknownNE,
          InvalidSlotAssignmentList, UnknownProfiles,
```

```
DuplicateUserLabel, AccessDenied, InvalidExternalTime,  
ProfileSuspended, ParameterViolation);"
```

b) *Remplacer:*

"L'entrée **administrationDomain** désigne le domaine auquel l'élément de réseau appartient."

*par:*

"L'entrée **registrationId** constitue la clé logique pour associer ces données de configuration à un élément de réseau installé. L'entrée **sRIInd** spécifie la nouvelle valeur pour la caractéristique de signalisation de l'état de la terminaison ONT ou de l'unité ONU. L'entrée **administrationDomain** désigne le domaine auquel appartient l'élément de réseau."

## 6.4 buildTCONT

Cette opération construit un TCONT dans le système de gestion par fournisseur. Les éléments ONT et ONU associés au TCONT doivent être configurés avant cette opération.

La signature de l'opération pour **buildTCONT** est représentée ci-dessous:

```
ManagedEntityIdType buildTCONT (  
    in ManagedEntityIdType nEId,  
    in TCONTType typeOFTCONT,  
    in BWType maxBW,  
    in BWType fixedBW,  
    in BWType guaranteedBW,  
    in UserLabelType userLabel)  
    raises (UnknownNE, AccessDenied, InsufficientPONBW,  
    UnsupportedTCONTType, DuplicateUserLabel);
```

L'entrée **nEId** identifie les éléments ONT et ONU où le TCONT doit se terminer. Le paramètre d'entrée **typeOFTCONT** identifie celui des cinq types de TCONT devant être construit. Les paramètres d'entrée **maxBW**, **fixedBW**, et **guaranteedBW** désignent les attributs de largeur de bande qui décrivent l'extrémité de liaison logique. L'entrée **userLabel** constitue une désignation d'opérateur unique pour le TCONT construit si une telle identification est souhaitée.

La valeur de retour du type **ManagedEntityIdType** fournit un identificateur unique identificateur pour le TCONT construit.

## 6.5 modifyTCONTParameters

Cette opération modifie les paramètres associés à un TCONT existant dans le système de gestion par fournisseur.

La signature de l'opération pour **modifyTCONTParameters** est présenté ci-dessous:

```
void modifyTCONTParameters (  
    in ManagedEntityIdType tCONTId,  
    in BWType maxBW,  
    in BWType fixedBW,  
    in BWType guaranteedBW,  
    in UserLabelType userLabel)  
    raises (UnknownManagedEntity, AccessDenied,  
    InsufficientPONBW, DuplicateUserLabel);
```

L'entrée **tCONTId** désigne le TCONT dont les caractéristiques doivent être modifiées. Les paramètres d'entrée **maxBW**, **fixedBW** et **guaranteedBW** désignent les nouveaux attributs de largeur de bande qui doivent caractériser l'extrémité de liaison logique. L'entrée **userLabel**

constitue une désignation d'opérateur unique pour le TCONT construit si une telle désignation est souhaitée.

La valeur de retour est de type **void**.

## 6.6 deleteTCONT

Cette opération supprime une configuration de TCONT du système de gestion par fournisseur. Par suite de cette opération, toutes les entités gérées automatiquement créées par suite de l'opération buildTCONT correspondante sont également supprimées. Un TCONT ne peut pas être supprimé s'il subsiste des connexions actives de sous-réseau qui lui sont associées.

La signature de l'opération pour **deleteTCONT** est présentée ci-dessous:

```
void deleteTCONT (
    in ManagedEntityIdType tCONTId)
    raises (UnknownManagedEntity, AccessDenied,
           RemainingSubnetworkConnections);
```

L'entrée **tCONTId** désigne le pont qui doit être supprimé.

La valeur de retour est de type **void**.

## 6.7 Modifications du § 9.2.1.20, "Exceptions"

a) *Remplacer:*

"L'exception **ParameterViolation** est déclenchée quand l'identificateur VPI est hors étendue ou en double."

*par:*

"L'exception **ParameterViolation** est déclenchée quand l'identificateur VPI est hors étendue ou en double ou quand la version des ONT et ONU en cours de fourniture ne peut pas prendre en charge l'option de signalisation de l'état spécifiée dans la demande."

b) *Ajouter ce qui suit en respectant l'ordre alphabétique:*

"L'exception **InsuffisantPONBW** est déclenchée lorsque le TCONT ne peut pas être construit ou modifié en raison d'une insuffisance de largeur de bande sur APONLink.

L'exception **UnsupportedTCONTType** est déclenchée si le type TCONT (1-5) spécifié dans l'opération n'est pas pris en charge par les ONT et ONU fournies."

## 7 Améliorations de l'interface de test

Dans le but de subdiviser et de cerner les problèmes d'offre des services téléphoniques et de données assurés par technologie BPON, l'interface de test Q834 a été améliorée par l'adjonction de trois nouveaux types de test: **metallicDropTest**, **maCLayerTest** et **drawDialtoneBreakTest**. Des versions programmées de ces types de test sont incluses. Comme il existe maintenant quatre types de tests programmés, la généralisation des tests **modifyResourceSelfTest** et **cancelResourceSelfTest** (appelés **modifyTestSchedule** et **cancelScheduledTest**) est proposée à titre de remplacement. Enfin, le § 9.15.1.7 a été reformulé pour souligner l'interprétation de **conductResourceSelfTest** pour assurer les tests de la fonctionnalité BORSHT d'une ONT ou d'une NT.

Toutes les modifications du § C.15 sont combinées avec d'autres améliorations de l'Annexe C contenues dans la Recommandation initiale et sont présentées dans le § 12.4.

## 7.1 metallicDropTest

Cette opération sert à effectuer un test **metallicDropTest** après l'identification d'une anomalie du système ou pour faire suite à une plainte d'un client au sujet du service.

La signature de l'opération pour le **metallicDropTest** est présentée ci-dessous:

```
DropTestResultsType metallicDropTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId)  
    raises (AccessDenied,  
           CommFailure,  
           UnknownManagedEntity);
```

Le paramètre d'entrée **testRequestorId** sert à identifier l'initiateur du test. Le paramètre d'entrée **port** désigne l'accès du service de téléphonie de l'interface UNI. Le paramètre d'entrée **serviceInstanceId** désigne l'instance de service l'instance de service associée à la demande de test.

La valeur de retour de **DropTestResultsType** donne des informations sur le test, à savoir le passage satisfaisant ou les résultats des mesures insatisfaisantes portant sur les points où le test a échoué.

## 7.2 scheduleMetallicDropTest

Cette opération sert à programmer un test **metallicDropTest**. Elle est utilisée par le système OMS pour établir de tels tests des accès du service téléphonique de manière régulière. Une condition préalable au lancement de cette opération est l'établissement d'un objet programmeur.

La signature de l'opération pour le **scheduleMetallicDropTest** est présentée ci-dessous:

```
TestTrackingObjectIdType scheduleMetallicDropTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId,  
    in UserLabelType schedulerName)  
    raises (AccessDenied,  
           UnknownManagedEntity,  
           UnknownScheduler,  
           InvalidScheduler);
```

Le paramètre d'entrée **testRequestorId** sert à identifier l'initiateur du test. Le paramètre d'entrée **port** désigne l'accès du service téléphonique de l'interface UNI. Le paramètre d'entrée **serviceInstanceId** désigne l'instance de service associée à la demande de test. Le paramètre d'entrée **schedulerName** sert à désigner le programmeur applicable à ce test.

La valeur de retour de **TestTrackingObjectIdType** désigne spécifiquement les tests programmés. L'objet de recherche de test existe tant qu'il n'est pas explicitement annulé au moyen de l'opération **cancelScheduledTest** ou jusqu'au moment où il atteint sa fin prévue.

Les résultats des tests sont consignés. Le type de donnée **DropTestResultsType** définit une partie des informations consignées.

## 7.3 mACLAYERTest

Cette opération sert à effectuer un test de couche MAC après l'identification d'une anomalie du système ou par suite d'une plainte de service d'un abonné. Ce test utilise la messagerie ARP diffusée pour détecter la présence d'équipements du client, mis en marche et connectés à un accès LAN port, une ONT ou une NT.

La signature de l'opération pour le **mACLAYERTest** est présentée ci-dessous:

```
short mACLAYERTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId)  
    raises (AccessDenied,  
           CommFailure,  
           UnknownManagedEntity);
```

Le paramètre d'entrée **testRequestorId** sert à identifier l'initiateur du test. Le paramètre d'entrée **port** désigne l'accès de service UNI LAN. Le paramètre d'entrée **serviceInstanceId** désigne l'instance de service associée à la demande de test.

La valeur de retour est de type **short** et donne le nombre de dispositifs détectés dans les installations des clients.

#### 7.4 **scheduleMACLayerTest**

Cette opération sert à programmer un test de couche MAC. Cette opération est utilisée par le système OMS pour établir l'exécution régulière de ce test des accès de service. Une condition préalable au lancement de cette opération est l'établissement d'un objet programmeur.

La signature de l'opération pour le **scheduleMACLayerTest** est présentée ci-dessous:

```
TestTrackingObjectIdType scheduleMACLayerTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId,  
    in UserLabelType schedulerName)  
    raises (AccessDenied,  
           UnknownManagedEntity,  
           UnknownScheduler,  
           InvalidScheduler);
```

Le paramètre d'entrée **testRequestorId** sert à identifier l'initiateur du test. Le paramètre d'entrée **port** désigne l'accès de service UNI LAN. Le paramètre d'entrée **serviceInstanceId** désigne l'instance de service associée à la demande de test. Le paramètre d'entrée **schedulerName** sert à désigner le programmeur applicable pour ce test.

La valeur de retour du type **TestTrackingObjectIdType** de manière exclusive désigne le test programmé. L'objet de recherche de test existe tant qu'il n'est pas explicitement annulé au moyen de l'opération **cancelScheduledTest** ou jusqu'au moment où il atteint sa fin prévue. Les résultats des tests sont consignés.

#### 7.5 **drawDialToneBreakTest**

Cette opération sert à effectuer un test relatif au service téléphonique après l'identification d'une anomalie du système ou par suite d'une plainte de service d'un abonné. Le test vérifie le trajet de signalisation entre une ONT ou une NT et un commutateur local pour une instance de service téléphonique.

La signature de l'opération pour le **drawDialToneBreakTest** est présentée ci-dessous:

```
boolean drawDialToneBreakTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId)  
    raises (AccessDenied,
```

```
CommFailure,  
UnknownManagedEntity);
```

Le paramètre d'entrée **testRequestorId** sert à identifier l'initiateur du test. Le paramètre d'entrée **port** désigne l'accès du service téléphonique de l'UNI. Le paramètre d'entrée **serviceInstanceId** désigne l'instance de service associée à la demande de test.

La valeur de retour, de type **boolean**, indique la réussite ou l'échec.

## 7.6 **scheduleDrawDialToneBreakTest**

Cette opération est utilisée par le système OMS pour établir un test de tonalité d'appel/d'interruption de tonalité d'appel pour des ports de service téléphonique à exécuter régulièrement. Une condition préalable au lancement de cette opération est l'établissement d'un objet programmeur.

La signature de l'opération pour le **scheduleDrawDialToneBreakTest** est présentée ci-dessous:

```
TestTrackingObjectIdType scheduleDrawDialToneBreakTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId,  
    in UserLabelType schedulerName)  
    raises (AccessDenied,  
           UnknownManagedEntity,  
           UnknownScheduler,  
           InvalidScheduler);
```

Le paramètre d'entrée **testRequestorId** sert à identifier l'initiateur du test. Le paramètre d'entrée **port** désigne l'accès de service téléphonique de l'interface UNI. Le paramètre d'entrée **serviceInstanceId** désigne l'instance de service associée à la demande de test. Le paramètre d'entrée **schedulerName** sert à désigner le programmeur applicable pour ce test.

La valeur de retour de **TestTrackingObjectIdType** désigne de manière exclusive le test programmé. L'objet de recherche de test existe tant qu'il n'est pas explicitement annulé au moyen de l'opération **cancelScheduledTest** ou jusqu'au moment où il atteint sa fin prévue. Les résultats des tests sont consignés.

## 7.7 **Modifications du § 9.15.1.5, "modifyResourceSelfTestSchedule"**

*Remplacer le § 9.1.15.5 par le suivant:*

### **"9.15.1.5 modifyTestSchedule**

Cette opération sert à modifier le programme d'exécution régulière d'un test. En cas de succès, le lancement du test est remplacé par l'itération suivante.

La signature de l'opération pour le **modifyTestSchedule** est présentée ci-dessous:

```
void modifyTestSchedule (  
    in TestTrackingObjectIdType testTrackingObjectId,  
    in UserLabelType newSchedulerName)  
    raises (UnknownTest, UnknownScheduler, InvalidScheduler,  
           AccessDenied);
```

Le paramètre d'entrée **testTrackingObjectId** sert à identifier le test programmé. Le paramètre d'entrée **newSchedulerName** sert à identifier le nouveau programme.

La valeur de retour est du type **void**."

## 7.8 Modifications du § 9.15.1.6, "cancelScheduledResourceSelfTest"

Remplacer le § 9.15.1.6 par le suivant:

### "9.15.1.6 cancelScheduledTest

Cette opération sert à annuler un test dont l'exécution est programmée. En cas de succès, cette opération annule le test avant son exécution et le remplace par l'heure de déclenchement suivante.

La signature de l'opération pour le **cancelScheduledTest** est présentée ci-dessous:

```
void cancelScheduledTest (  
    in TestTrackingObjectIdType testTrackingObjectId)  
    raises (UnknownTest, UncontrolledTestInProgress, AccessDenied);
```

Le paramètre d'entrée **testTrackingObjectId** sert à identifier le test programmé auquel il faut mettre fin.

La valeur de retour est du type **void**."

## 7.9 Modifications du § 9.15.1.7, "conductResourceSelfTest"

Remplacer:

"Cette opération sert à lancer un essai d'autovérification d'une ressource après l'identification d'une panne de système ou une réclamation concernant un service d'abonné."

par:

"Cette opération sert à lancer un test automatique d'une ressource après la détection d'une anomalie du système ou après une plainte concernant un service d'abonné. Le test automatique de la ressource peut servir à tester la fonctionnalité BORSHT dans une ONT ou une NT."

## 8 Améliorations de l'interface fournisseur d'événement

Pour informer le système de gestion par l'opérateur de la fin de l'exécution d'activités de gestion programmées ou de longue durée demandées par le système de gestion par l'opérateur, une autre interface et le mappage avec l'objet événement structuré ont été créés. Pour l'heure, la principale utilisation de cette interface est la notification de la fin des activités de gestion du logiciel de NE.

Toutes les modifications associées du § C.6 sont combinées avec d'autres améliorations de l'Annexe C de la Recommandation initiale et sont présentées dans le § 12.3.

### 8.1 ActivityCompletionEventSupplier

L'interface a pour objet d'annoncer au système de gestion par l'opérateur via le service de notification du groupe OMG des événements relatifs à l'exécution des opérations de gestion programmées ou de longue durée demandées par le système de gestion par l'opérateur au système de gestion par le fournisseur. Cette interface ne comporte pas d'opération. Toutefois, elle assure le mappage de l'en-tête fixe et les mappages des données filtrables pour l'objet événement structuré utilisé pour conduire l'information d'événement via le conduit d'événement du service de notification du groupe OMG.

Dans l'en-tête fixe, le **domain\_type** est mis à "telecommunications", le **type\_name** à "ActivityCompletion" et le **event\_name** à une chaîne constante qui a l'une des valeurs suivantes: "SoftwareDownload", "SoftwareCommit" ou "SoftwareActivation".<sup>1</sup>

---

<sup>1</sup> Pour l'heure, cette interface traite uniquement des activités faisant intervenir le téléchargement de logiciels vers les éléments de réseau.

Le mappage avec les données filtrables est constitué d'une paire de composantes. La première est un identificateur de chaîne pour un nom de donnée, le second est la valeur pour cet élément de donnée. Les identificateurs de chaîne sont des constantes définies dans cette interface. Par ailleurs, les paires de données filtrables doivent se présenter dans un ordre spécifique.

L'ordre des éléments filtrables pour un event\_name de "SoftwareDownload", "SoftwareCommit" ou "SoftwareActivation" se présente comme suit:

- EventTime;
- NotificationIdentifier;
- CorrelatedNotifications;
- TrackingObjectId;
- SuccessIndication;
- AdditionalText.

La valeur de EventTime a une syntaxe de type GeneralizedTimeType; elle désigne l'instant auquel l'activité logicielle dans l'élément de réseau s'est terminée.

La valeur NotificationIdentifier a une syntaxe de type NotificationIdentifierType; elle fournit un numéro de séquence de référence pour l'événement. La valeur de CorrelatedNotifications a une syntaxe de type CorrelatedNotificationType; elle fournit une liste de numéros de référence pour d'autres notifications d'événement fournies par le système de gestion par le fournisseur pour les activités logicielles. En l'absence de notifications connexes, c'est la valeur de l'ensemble vide qui est fournie.

La valeur de TrackingObjectId a une syntaxe TrackingObjectIdType; elle désigne l'activité de gestion programmée ou de longue durée demandée par le système de gestion par l'opérateur. Cet identificateur a été fourni par le système de gestion par le fournisseur au moment où la demande a été faite. La valeur de SuccessIndication a une syntaxe booléenne; elle indique si l'activité est ou n'est pas terminée.

Enfin, la valeur de AdditionalText a une chaîne de syntaxe. Cet élément de donnée fournit un emplacement pour passer toute information textuelle du système de gestion par le fournisseur relatif à un échec de l'activité. En l'absence d'information additionnelle, c'est une chaîne zéro qui sera transférée.

## 8.2 Tableau 1, "q834\_4 module organisation"

Remplacer la ligne C.6 par la suivante:

"

C.6	Q834EventPublisher	AlarmEventSupplier, SecurityEventSupplier, DiscoveryEventSupplier ActivityCompletionSupplier	Publication d'événements
-----	--------------------	---	--------------------------

"

## 9 Améliorations de l'interface Q.834 Common pour la gestion des capacités

L'interface Q834 Common est améliorée par l'adjonction d'un enregistrement de ressources OLT et de plusieurs enregistrements de ressources d'accès. Ces types d'enregistrement sont nécessaires pour assister le transfert d'informations relatives à l'utilisation des ressources physiques et logiques associées aux éléments de réseau BPON. Ces éléments sont appelés OLTResourceRecordType, ATMPortResourceRecordType, TDMPhysicalPortResourceRecordType et

EthernetPhysicalPortResourceRecordType. Le seul endroit où ces adjonctions ont été faites dans la Recommandation initiale est le § C.3.

Toutes les modifications du § C.3 sont combinées avec d'autres améliorations que l'on trouve dans la Recommandation initiale et sont présentées dans le § 12.2.

## 10 Améliorations de l'Annexe A "Dictionnaire de données"

Ajouter les éléments suivants dans le Tableau A.1, Eléments de données et définitions:

"

BWType	Cet attribut désigne la quantité de largeur de bande attribuée.	flottante	
DBAFairnessPMHistoryDataType	Indique la structure de l'enregistrement correspondant aux données de performance collectées sur 15 minutes pour un point de surveillance TC Adaptor à un accès PON d'une terminaison OLT.	struct	
DropTestResultsType	Cet élément de données fournit les résultats du test suivant la fin d'un metallic drop test.	struct	
RegistrationIdType	Cet élément de données fournit la clé de corrélation utilisée pour construire l'association entre les connexions de service préalablement fournies et un NE qui doit encore être construit.	chaîne	
TCONTTEPMHistoryDataType	Indique la structure de l'enregistrement correspondant aux données de performance collectées sur 15 minutes pour un point de surveillance TCONT Buffer à un accès PON d'une terminaison OLT.	struct	
TCONTType	Cet attribut spécifie le type de TCONT construit.	courte	Les valeurs permises sont 1, 2, 3, 4, 5

"

## 11 Améliorations de l'Annexe B, "Exceptions"

Ajouter, par ordre alphabétique, l'élément suivant dans le Tableau B.1, Exceptions:

UnsupportedTCONTType	Cette exception est déclenchée quand le type TCONT spécifié dans l'opération n'est pas pris en charge par les éléments ONT et ONU spécifiés.
----------------------	--

## 12 Améliorations de l'Annexe C, "Fichiers en langage IDL"

### 12.1 Améliorations du § C.2, "Q834Build.idl"

#### 12.1.1 Imports, typedefs et exceptions

Ajouter les imports, typedefs et exceptions suivants:

```
"typedef Q834Common::RegistrationIdType RegistrationIdType;
```

```

#define InsufficientPONBW Q834Common::InsufficientPONBW;
#define UnknownNE Q834Common::UnknownNE;
typedef float BWType;
typedef short TCONTType;
exception UnsupportedTCONTType {};

```

### 12.1.2 Interface Builder

#### a) *Remplacer:*

"/" Voir § 9.2.1.1 pour la description du comportement de cette opération

```

ManagedEntityIdType buildNode (
    in NEKindType nEKind,
    in string supplierName,
    in string location,
    in VersionType hWVersion,
    in SerialNumType serialNum,
    in NameSeqType alarmSeverityProfiles,
    in NameSeqType thresholdDataProfiles,
    in SlotAssignmentSeqType slotAssignmentList,
    in ManagedEntityIdType port, // OLT PON port
    in string modelCode,
    in string systemTitle,
    in VersionSeqType softwareVersions,
    in UserLabelType nEUserLabel,
    in ExternalTimeType externalTime,
    in SystemTimingType systemTiming,
    in AdministrationDomainType administrationDomain )
raises (UnrecognisedVersion,
        InvalidSerialNumSyntax,
        DuplicateSerialNumber,
        UnknownProfiles,
        UnknownManagedEntity,
        DuplicateUserLabel,
        AccessDenied,
        InvalidExternalTime,
        UnknownSystemTimingSource,
        ProfileSuspended);

```

*par:*

"/" Voir § 9.2.1.1 pour la description du comportement de cette opération

```

ManagedEntityIdType buildNode (
    in NEKindType nEKind,
    in string supplierName,
    in string location,
    in VersionType hWVersion,
    in SerialNumType serialNum,
    in NameSeqType alarmSeverityProfiles,
    in NameSeqType thresholdDataProfiles,
    in SlotAssignmentSeqType slotAssignmentList,
    in ManagedEntityIdType port, // OLT PON port
    in string modelCode,
    in string systemTitle,
    in VersionSeqType softwareVersions,
    in UserLabelType nEUserLabel,
    in ExternalTimeType externalTime,
    in SystemTimingType systemTiming,
    in RegistrationIdType registrationId,
    in boolean sRInd,

```

```

in AdministrationDomainType administrationDomain )
raises (UnrecognisedVersion,
        InvalidSerialNumSyntax,
        DuplicateSerialNumber,
        UnknownProfiles,
        ParameterViolation
        UnknownManagedEntity,
        DuplicateUserLabel,
        AccessDenied,
        InvalidExternalTime,
        UnknownSystemTimingSource,
        ProfileSuspended,
        ParameterViolation);"

```

b) *Remplacer:*

"/" Voir § 9.2.1.2 pour la description du comportement de cette opération

```

void assignUserLabelsToNE (
in SerialNumType serialNum,
in UserLabelType nEUserLabel,
in AdministrationDomainType administrationDomain)
raises (InvalidSerialNumSyntax,
        DuplicateSerialNumber,
        DuplicateUserLabel,
        AccessDenied);"

```

*par:*

"/" voir § 9.2.1.2 pour la description du comportement de cette opération

```

void assignUserLabelsToNE (
in SerialNumType serialNum,
in UserLabelType nEUserLabel,
in RegistrationIdType registrationId,
in AdministrationDomainType administrationDomain)
raises (InvalidSerialNumSyntax,
        DuplicateSerialNumber,
        DuplicateUserLabel,
        AccessDenied);"

```

c) *Remplacer:*

"/" Voir 9.2.1.3 pour la description du comportement de cette opération

```

void modifyNode (
in ManagedEntityIdType managedEntityId,
in SlotAssignmentSeqType newSlotAssignmentList,
in NameType newAlarmSeverityProfiles,
in NameSeqType newThresholdDataProfiles,
in ManagedEntityIdType port, // OLT PON Port
in string newModelCode,
in UserLabelType newNEuserLabel,
in ExternalTimeType externalTime,

```

```

in AdministrationDomainType administrationDomain )
raises (UnknownManagedEntity,
        UnknownNE,
        InvalidSlotAssignmentList,
        UnknownProfiles,
        DuplicateUserLabel,
        AccessDenied,
        InvalidExternalTime,
        ProfileSuspended);"

```

*par:*

" // Voir § 9.2.1.3 pour la description du comportement de cette opération

```

void modifyNode (
    in ManagedEntityIdType managedEntityId,
    in SlotAssignmentSeqType newSlotAssignmentList,
    in NameSeqType newAlarmSeverityProfiles,
    in NameSeqType newThresholdDataProfiles,
    in ManagedEntityIdType port, // OLT PON Port
    in string newModelCode,
    in UserLabelType newNEuserLabel,
    in ExternalTimeType externalTime,
    in RegistrationIdType registrationId,
    in boolean sRInd,
    in AdministrationDomainType administrationDomain )
raises (UnknownManagedEntity,
        UnknownNE,
        InvalidSlotAssignmentList,
        UnknownProfiles,
        DuplicateUserLabel,
        AccessDenied,
        InvalidExternalTime,
        ProfileSuspended,
        ParameterViolation);"

```

d) *Ajouter les opérations suivantes:*

"// Voir § 9.2.1.20 pour la description du comportement de cette opération

```

ManagedEntityIdType buildTCONT (
    in ManagedEntityIdType nEId,
    in TCONTType typeOFTCONT,
    in BWType maxBW,
    in BWType fixedBW,
    in BWType guaranteedBW,
    in UserLabelType userLabel)
raises (UnknownNE, AccessDenied, InsufficientPONBW,
        UnsupportedTCONTType, DuplicateUserLabel);

```

// Voir § 9.2.1.21 pour la description du comportement de cette opération

```

void modifyTCONTParameters (
    in ManagedEntityIdType tCONTId,
    in BWType maxBW,
    in BWType fixedBW,
    in BWType guaranteedBW,
    in UserLabelType userLabel)
raises (UnknownManagedEntity, AccessDenied,
        InsufficientPONBW, DuplicateUserLabel);

```

// Voir § 9.2.1.22 pour la description du comportement de cette opération

```
void deleteTCONT (
    in ManagedEntityIdType tCONTId)
    raises (UnknownManagedEntity, AccessDenied,
           RemainingSubnetworkConnections);"
```

## 12.2 Améliorations du § C.3, "Q834Common.idl"

### 12.2.1 Structures et Typedefs

a) *Ajouter les typedefs et structures suivants:*

```
"typedef boolean SRIndType;
```

```
"typedef boolean SRIndType;
```

```
typedef string RegistrationIdType;
```

```
struct DBAFairnessPMHistoryDataType {
    long long recordId;
    ManagedEntityIdType monitoringPoint;
    GeneralizedTimeType periodEndTime;
    boolean suspectIntervalFlag;
    NameType thresholdDataId;
    float type2Variance; //value = -1 if type 2 is unsupported by NE
    float type3Variance; //value = -1 if type 3 is unsupported by NE
    float type4Variance; //value = -1 if type 4 is unsupported by NE
    float type5Variance; //value = -1 if type 5 is unsupported by NE
};
```

```
struct TCONTTEPMHistoryDataType {
    long long recordId;
    ManagedEntityIdType monitoringPoint;
    GeneralizedTimeType periodEndTime;
    boolean suspectIntervalFlag;
    NameType thresholdDataId;
    float averageReceive_AssignRate;
    float maximumReceive_AssignRate;
    float minimumReceive_AssignRate;
};
```

```
struct OLTResourceRecordType {
    long long recordId;
    ManagedEntityIdType oLTResourceId;
    ManagedEntityIdType containingNEId;
    GeneralizedTimeType collectionTimestamp;
    SlotAssignmentSeqType slotAssignmentList;
};
```

```
struct ATMPortResourceRecordType {
    long long recordId;
    ManagedEntityIdType portResourceId;
    ManagedEntityIdType portId;
    GeneralizedTimeType collectionTimestamp;
    unsigned long maxVPConnectionCount;
    unsigned long maxVCConnectionCount; //Zero if APON port
    unsigned long reservedVPConnectionCount;
    unsigned long reservedVCConnectionCount; //Zero if APON
    port
    unsigned long assignedVPConnectionCount;
    unsigned long assignedVCConnectionCount; //Zero if APON
    port
};
```

```

float      maxBW;
float      reservedBW;
float      assignedBW;
}; //PON ports in BPON support ATM

```

```

struct TDMPPhysicalPortResourceRecordType {
    long long recordId;
    ManagedEntityIdType    portResourceId;
    ManagedEntityIdType    portId;
    GeneralizedTimeType    collectionTimestamp;
    unsigned long          maxTSCCount;
    unsigned long          reservedTSCCount;
    unsigned long          assignedTSCCount;
};

```

```

struct EthernetPhysicalPortResourceRecordType {
    long long recordId;
    ManagedEntityIdType    portResourceId;
    ManagedEntityIdType    portId;
    GeneralizedTimeType    collectionTimestamp;
    unsigned long          maxVLANTagCount;
    unsigned long          reservedVLANTagCount;
    unsigned long          assignedVLANTagCount;
};"

```

b) *Replacer:*

```

"struct    ONTType    {
    NEFSANType    nEFSAN;
    ManagedEntityIdType    upstreamNEFSAN;
};"

```

*par:*

```

"struct    ONTType    {
    NEFSANType    nEFSAN;
    ManagedEntityIdType    upstreamNEFSAN;
    SRIndType    sRInd;
    short    maxDataGrants;
    RegistrationIdType    registrationId;
};"

```

c) *Replacer:*

```

"struct    ONUType    {
    NEFSANType    nEFSAN;
    ManagedEntityIdType    upstreamNEFSAN;
    ManagedEntityIdSeqType    subtendingNEFSANList;
};"

```

*par:*

```

"struct    ONUType    {
    NEFSANType    nEFSAN;
    ManagedEntityIdType    upstreamNEFSAN;
    ManagedEntityIdSeqType    subtendingNEFSANList;
    SRIndType    sRInd;
    short    maxDataGrants;
    RegistrationIdType    registrationId;
};"

```

d) *Remplacer:*

```
"struct      PlugInUnitFType  {
    ManagedEntityIdType      plugInUnitFId;
    ManagedEntityIdType      containingNEId;
    EquipmentHolderAddressType  containingSlotAddress;
    AdministrativeStateType  administrativeState;
    AvailabilityStatusSetType  availabilityStatus;
    OperationalStateType      operationalState;
    string                    equipmentCode;
    string                    functionCode;
    string                    supplierName;
    VersionType               hardwareVersion;
    string                    serialNumber;
    short                    portCount;
    NameSeqType               alarmSeverityAssignmentProfileNames;
    NameSeqType               thresholdDataNames;
    UserLabelType             circuitPackUserLabel;
    ManagedEntityIdSeqType    supportedByManagedEntityList;
};"
```

*par:*

```
"struct      PlugInUnitFType  {
    ManagedEntityIdType      plugInUnitFId;
    ManagedEntityIdType      containingNEId;
    EquipmentHolderAddressType  containingSlotAddress;
    AdministrativeStateType  administrativeState;
    AvailabilityStatusSetType  availabilityStatus;
    OperationalStateType      operationalState;
    string                    equipmentCode;
    string                    functionCode;
    string                    supplierName;
    VersionType               hardwareVersion;
    VersionSeqType            softwareVersions;
    string                    serialNumber;
    short                    portCount;
    NameSeqType               alarmSeverityAssignmentProfileNames;
    NameSeqType               thresholdDataNames;
    UserLabelType             circuitPackUserLabel;
    ManagedEntityIdSeqType    supportedByManagedEntityList;
};"
```

### 12.2.2 Interface ProbableCause

*Dans interface ProbableCause, ajouter les définitions suivantes:*

```
"const unsigned short LOSS_OF_MINISLOT = 35;
const unsigned short STATUS_REPORTING_HANDSHAKE_FAILURE = 36;"
```

### 12.2.3 Interface PMCategory

*Dans interface PMCategory, ajouter les définitions suivantes:*

```
"const unsigned short DBA_FAIRNESS_PM = 24;
const unsigned short TCONT_TRAFFIC_PM = 25;"
```

## 12.2.4 Interface MonitoringParameter

*Dans interface MonitoringParameter, ajouter les définitions suivantes:*

```
"const      string type2Var = "Type2Var";
const      string type3Var = "Type3Var";
const      string type4Var = "Type4Var";
const      string type5Var = "Type5Var";
const      string averageRec_AssignRate = "AverageRec_AssignRate";
const      string maximumRec_AssignRate = "MaximumRec_AssignRate";
const      string minimumRec_AssignRate = "MinimumRec_AssignRate";"
```

## 12.2.5 Interface RecordSetType

*Dans interface RecordSetType, ajouter les définitions suivantes:*

```
"const unsigned short DBAFAIRNESSPMHISTORYDATA = 19;
const unsigned short TCONTTEPMHISTORYDATA = 20;"
const unsigned short OLTRESOURCERECORD = 21;
const unsigned short ATMPORTRESOURCERECORD = 22;
const unsigned short TDMPHYSICALPORTRESOURCERECORD = 23;
const unsigned short ETHERNETPORTRESOURCERECORD = 24;"
```

## 12.3 Améliorations du § C.6, "Q834Eventpublisher.idl"

### 12.3.1 Importations

*Ajouter les importations suivantes:*

```
"typedef Q834Common::TrackingObjectIdType TrackingObjectIdType"
```

### 12.3.2 ActivityCompletionEventSupplier

*Ajouter la nouvelle interface suivante:*

```
"Mappages d'interfaces ActivityCompletionEventSupplier :
itut_x780::ManagedObject {

/* Mappages d'événements structurés dans un en-tête:
domain_type is set to "telecommunications",
type_name is set to "ActivityEvent", and
event_name is set to one of the following constant strings
provided below.
*/
const string softwareDownload = "SoftwareDownload";
const string softwareCommit = "SoftwareCommit";
const string softwareActivation = "SoftwareActivation";

/* Des éléments additionnels à mapper dans le § des données filtrables de
l'objet événement structuré sont donnés ci-dessous.
*/
const string eventTime = "EventTime";
const string additionalText = "AdditionalText";
const string notificationIdentifier = "NotificationIdentifier";
const string correlatedNotifications = "CorrelatedNotifications";
const string successIndication = "SuccessIndication";
const string trackingObjectId = "TrackingObjectId";

/*
le mappage avec des données filtrables dans l'événement structuré est donné
ci-dessous pour un événement d'activité logicielle.

{
```

```

        {"EventTime", any (GeneralizedTimeType)},
        {"NotificationIdentifier", any (NotificationIdentifierType)},
        {"CorrelatedNotifications", any (CorrelatedNotificationType)},
        {"TrackingObjectId", any (TrackingObjectIdType)},
        {"SuccessIndication", boolean},
        {"AdditionalText", any (string)}
    }
*/

}; // interface ActivityCompletionEventSupplier"

```

## 12.4 Améliorations du § C.15, "Q834Test.idl"

### 12.4.1 Structures et Typedefs

*Ajouter les typedefs et structures suivants:*

```

"typedef float HazardousPotentialType; //in volts

struct ForElectroMotiveForceType {
    float acVoltageTipToGround;
    float acVoltageRingToGround;
    float dcVoltageTipToGround;
    float dcVoltageRingToGround;
}; //in volts

struct ResistiveFaultType {
    float dcResistanceTipToRing;
    float dcResistanceTipToGround;
    float dcResistanceRingToGround;
}; //in volts

struct ReceiverOffHookType {
    float dcResistance1TipToRing;
    float dcResistance2TipToRing;
}; //in ohms

struct ReceiverOffHookType {
    float dcResistance1TipToRing;
    float dcResistance2TipToRing;
}; //in ohms

struct PresenceOfRingerType {
    float acImpedanceTipToRing;
    float acImpedanceTipToGround;
    float acImpedanceRingToGround;
}; //in ohms

struct NetworkTerm1DCSignatureType {
    float dcVoltage1TipToRing;
    float dcVoltage2TipToRing;
};

struct DropTestResultsType {
    short resultsMask;
    HazardousPotentialType hazardousPotential;
    ForElectroMotiveForceType foreignElectroMotiveForce;
    ResistiveFaultType resistiveFault;
    ReceiverOffHookType receiverOffHook;
    PresenceOfRingerType ringer;
    boolean networkTermination1dcSignatureTest;
};

```

```
/* ResultsMask est un entier de 0 à 6, où 0 indique que tous les tests ont été
passés, un entier différent de zéro indique que les mesures correspondant au
test ayant échoué sont en cours d'envoi, où 1 indique une défaillance au niveau
du test de potentiel de risque, 2 indique l'échec du test de force
électromotrice étrangère, 3 indique l'échec du test du défaut de résistivité,
4 indique l'échec du test de combiné décroché, 5 indique l'échec du test de la
présence de la sonnerie et 6 indique l'échec du test de signature de la
terminaison 1 DC de réseau. */
```

## 12.4.2 TestActionPerformer Interface

### a) *Remplacer:*

```
/// Voir § 9.15.1.5 pour la description du comportement de cette opération
```

```
void modifyResourceSelfTestSchedule(
    in TestTrackingObjectIdType testTrackingObjectId,
    in UserLabelType newSchedulerName)
    raises (AccessDenied,
           UnknownTest,
           UnknownScheduler,
           InvalidScheduler);
```

```
/// Voir § 9.15.1.6 pour la description du comportement de cette opération
```

```
void cancelScheduledResourceSelfTest (
    in TestTrackingObjectIdType testTrackingObjectId)
    raises (AccessDenied,
           UnknownTest);"
```

*par:*

```
/// Voir § 9.15.1.5 pour la description du comportement de cette opération
```

```
void modifyTestSchedule (
    in TestTrackingObjectIdType testTrackingObjectId,
    in UserLabelType newSchedulerName)
    raises (UnknownTest,
           UnknownScheduler,
           InvalidScheduler,
           AccessDenied);
```

```
/// Voir § 9.15.1.6 pour la description du comportement de cette opération
```

```
void cancelScheduledTest (
    in TestTrackingObjectIdType testTrackingObjectId)
    raises (UnknownTest, UncontrolledTestInProgress,
           AccessDenied);"
```

### b) *Ajouter les opérations suivantes:*

```
/// Voir § 9.15.1.17 pour la description du comportement de cette opération
```

```
DropTestResultsType metallicDropTest (
    in UserIdType testRequestorId,
    in ManagedEntityIdType port,
    in ServiceInstanceIdType serviceInstanceId)
```

```
        raises (AccessDenied,
               CommFailure,
               UnknownManagedEntity);
```

```
// Voir § 9.15.1.18 pour la description du comportement de cette opération
```

```
TestTrackingObjectIdType scheduleMetallicDropTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId,  
    in UserLabelType schedulerName)  
    raises (AccessDenied,  
           UnknownManagedEntity,  
           UnknownScheduler,  
           InvalidScheduler);
```

```
// Voir § 9.15.1.19 pour la description du comportement de cette opération
```

```
short mACLAYERTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId)  
    raises (AccessDenied,  
           CommFailure,  
           UnknownManagedEntity);
```

```
// Voir § 9.15.1.20 pour la description du comportement de cette opération
```

```
TestTrackingObjectIdType scheduleMACLayerTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId,  
    in UserLabelType schedulerName)  
    raises (AccessDenied,  
           UnknownManagedEntity,  
           UnknownScheduler,  
           InvalidScheduler);
```

```
// Voir § 9.15.1.21 pour la description du comportement de cette opération
```

```
boolean drawDialToneBreakTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId)  
    raises (AccessDenied,  
           CommFailure,  
           UnknownManagedEntity);
```

```
// Voir § 9.15.1.22 pour la description du comportement de cette opération
```

```
TestTrackingObjectIdType scheduleDrawDialToneBreakTest (  
    in UserIdType testRequestorId,  
    in ManagedEntityIdType port,  
    in ServiceInstanceIdType serviceInstanceId,  
    in UserLabelType schedulerName)  
    raises (AccessDenied,  
           UnknownManagedEntity,  
           UnknownScheduler,  
           InvalidScheduler);"
```

### 13 Améliorations de l'Annexe D, "Exemple de gabarits d'extrémité"

a) *Après le Tableau D.2, points d'extrémité de Port UNI, ajouter l'alinéa suivant:*

"Pour gérer la fourniture de la connexion de service pour les systèmes BPON assurant l'attribution dynamique de largeur de bande, tous les exemples sous l'en-tête de colonne "Paramètre" dans le Tableau D.2 pourraient être complétés pour y inclure le ManagedEntityIdType du TCONT auquel appartient la connexion de service. La désignation TCONT apparaîtrait en premier. Cela est uniquement requis au cas où la fourniture de T-CONT est expressément prise en charge."

b) *A la fin de l'Annexe D, Exemples de gabarits de point d'extrémité, ajouter:*

"Le Tableau D.3 illustre des exemples de type de connexion de sous-réseau qui sont formés via l'utilisation de la connexion provisionConnection de l'opération spécifiée dans l'interface Q834ServiceProvisioning::ServiceProvisioner."

**Tableau D.3/Q.834.4 – Types de connexion de sous-réseau**

Service	Point d'extrémité A	Point d'extrémité Z	Connexion de sous-réseau
DS1	TDM DS3	TDM DS1	ds1SubnetworkConnection
DS1	ATM DS3 ou OCn	TDM DS1	vcSubnetworkConnection
DS1	TDM DS3	TDM DS3	ds1SubnetworkConnection
DS3	TDM DS3	TDM DS3	ds3SubnetworkConnection
DS3	ATM DS3 ou OCn	TDM DS3	vcSubnetworkConnection
Voice	ATM DS3 ou OCn	RJ-11	vcSubnetworkConnection
Voice	TDM DS1	RJ-11	ds0SubnetworkConnection
Bridged LAN	ATM DS3 ou OCn	Ethernet	vcSubnetworkConnection
Bridged LAN	Ethernet	Ethernet	vcSubnetworkConnection



## SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
<b>Série Q</b>	<b>Commutation et signalisation</b>
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de nouvelle génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication