



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**X.780**

**Corrigendum 1**  
(10/2001)

SÉRIE X: RÉSEAUX DE DONNÉES ET  
COMMUNICATION ENTRE SYSTÈMES OUVERTS  
Gestion OSI – Fonctions de gestion et fonctions ODMA

---

Directives concernant le RGT pour la définition  
d'objets gérés CORBA

**Corrigendum 1**

Recommandation UIT-T X.780 (2001) – Corrigendum 1

(Antérieurement Recommandation du CCITT)

---

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

<b>RÉSEAUX PUBLICS DE DONNÉES</b>	
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
<b>INTERCONNEXION DES SYSTÈMES OUVERTS</b>	
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 des couches	X.280–X.289
Tests de conformité	X.290–X.299
<b>INTERFONCTIONNEMENT DES RÉSEAUX</b>	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.399
<b>SYSTÈMES DE MESSAGERIE</b>	<b>X.400–X.499</b>
<b>ANNUAIRE</b>	<b>X.500–X.599</b>
<b>RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES</b>	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
<b>GESTION OSI</b>	
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
<b>Fonctions de gestion et fonctions ODMA</b>	<b>X.730–X.799</b>
<b>SÉCURITÉ</b>	<b>X.800–X.849</b>
<b>APPLICATIONS OSI</b>	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
<b>TRAITEMENT RÉPARTI OUVERT</b>	<b>X.900–X.999</b>

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

## **Recommandation UIT-T X.780**

### **Directives concernant le RGT pour la définition d'objets gérés CORBA**

#### **CORRIGENDUM 1**

#### **Source**

Le Corrigendum 1 de la Recommandation X.780 (2001) de l'UIT-T, élaboré par la Commission d'études 4 (2001-2004) de l'UIT-T, a été approuvé le 7 octobre 2001 selon la procédure définie dans la Résolution 1 de l'AMNT.

## 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.

## 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 2002

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.

## TABLE DES MATIÈRES

	<b>Page</b>
1) Paragraphe 2.1 .....	1
2) Paragraphe 5.1.3 .....	1
3) Paragraphe 5.5 .....	1
4) Paragraphe 6.3 .....	1
5) Nouveau paragraphe 6.3.6 .....	2
6.3.6 Types communs .....	2
6) Paragraphe 6.4 .....	2
7) Paragraphe 6.6 .....	2
8) Paragraphe 6.9.1 .....	2
6.9.1 Opérations de création .....	2
9) Paragraphe 6.9.1.2 .....	3
10) Nouveau paragraphe 6.9.1.3 .....	3
11) Paragraphe 6.9.1.4 .....	3
12) Paragraphe 9.1 .....	3
13) Annexe A .....	3
14) Nouvelle Annexe C .....	5
Annexe C – Module IDL des types communs en architecture Corba .....	5



## Recommandation UIT-T X.780

### Directives concernant le RGT pour la définition d'objets gérés CORBA

#### CORRIGENDUM 1

#### 1) *Paragraphe 2.1*

*Ajouter une nouvelle référence:*

- [8] UIT-T X.739 (1993) | ISO/CEI 10164-11 (1994); *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion- systèmes: objets et attributs métriques.*

#### 2) *Paragraphe 5.1.3*

*Après la deuxième phrase du premier alinéa, ajouter la phrase suivante:*

Une chaîne de caractères est définie pour chaque chaîne nominative de paquetage conditionnel.

#### 3) *Paragraphe 5.5*

*Dans le langage IDL, supprimer les lignes suivantes:*

```
valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};

valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};
```

*et les remplacer par les suivantes:*

```
valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};

valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};
```

#### 4) *Paragraphe 6.3*

*Supprimer la dernière phrase du premier alinéa:*

Dans des versions futures de CORBA, on prévoit d'autoriser les anomalies définies par l'utilisateur pour l'accès aux attributs et les directives définies ici pourront être modifiées afin d'en tirer parti.

*Après le premier alinéa, ajouter l'alinéa suivant:*

La définition d'attributs contenant des références à des objets gérés devrait être évitée. Il est préférable d'utiliser les noms de ces objets gérés.

## 5) *Nouveau paragraphe 6.3.6*

*Ajouter un nouveau paragraphe 6.3.6 intitulé "Types communs" comme suit:*

### 6.3.6 Types communs

On trouvera dans l'Annexe C un module définissant des types communs qui sont utilisés par plusieurs Recommandations. Ce module vise à offrir un lieu unique pour la définition de ces types. D'autres types seront au besoin ajoutés à ce module.

## 6) *Paragraphe 6.4*

*Au premier alinéa, après la deuxième phrase, ajouter la phrase suivante:*

Il convient cependant d'éviter d'utiliser l'utilisation de paramètres contenant des références à des objets gérés. Il est préférable d'utiliser les noms de ces objets gérés.

## 7) *Paragraphe 6.6*

*Après le troisième alinéa, ajouter l'alinéa suivant:*

Par ailleurs, pour chaque paquetage conditionnel défini, l'on doit définir une chaîne de caractères pouvant être utilisée dans l'attribut des paquetages pour les instances d'objet géré utilisant ce paquetage conditionnel. Cette chaîne doit être entièrement précisée afin de différencier les paquetages d'autres fichiers IDL. Par exemple:

```
const string administrativeStatePackage =  
    "itut_m3120::administrativeStatePackage";
```

## 8) *Paragraphe 6.9.1*

*Remplacer le paragraphe 6.9.1 par ce qui suit:*

### 6.9.1 Opérations de création

Chaque interface de fabrique doit définir une opération unique à utiliser par les clients pour créer des objets. Cette opération doit avoir pour nom "create" (création) et doit renvoyer une référence de type *ManagedObject*. Cette référence doit être utilisée pour renvoyer une référence à l'objet qui vient d'être créé. Cette référence peut être "centrée" par le client sur le type d'interface propre à cet objet ou être utilisée directement pour les opérations de classe de base. Les cinq premiers paramètres de chaque opération de création sont toujours les mêmes. Viennent ensuite des paramètres pour chaque attribut fixé par création ou pouvant être modifié qui est défini pour l'objet géré. (Un attribut fixé par création est un attribut pour lequel l'objet n'a pas d'opération "set" (écriture), mais pour lequel une valeur est spécifiée dans l'opération de création.) Les noms de ces paramètres sont identiques à celui de l'attribut. (Il s'agit du nom d'une opération d'accès à l'attribut moins le suffixe "Get" ou "Set".) Chaque opération de création doit aussi accepter des paramètres permettant de fixer les valeurs de n'importe quel attribut fixé par création ou pouvant être modifié de toutes les superclasses de l'objet créé par la fabrique. Donnons un exemple d'opération de création pour une fabrique d'équipement:

```
ManagedObject create(  
    in NameBindingType nameBinding, // module name containing Name Binding  
                                   // info.  
    in MOnameType superiorObject, // Name of containing object.  
    in string reqID,               // Requested ID value for name, will be  
                                   // empty if auto-naming is to be used.  
    out MOnameType name,           // Entire name of newly created object.  
    in StringSetType packages,     // List of packages requested.  
    ...                             // Writeable and set-by-create values  
                                   // for Equipment superclass attributes.
```

```
... // Writeable and set-by-create values
// for Equipment attributes.
);
```

### 9) **Paragraphe 6.9.1.2**

*Dans la première phrase du premier alinéa, remplacer les mots "est une référence à" par "est le nom de". Dans la troisième phrase du premier alinéa, remplacer les mots "la référence fournie" par "l'objet supérieur". Dans la dernière phrase du deuxième alinéa, remplacer les mots "la référence à l'objet supérieur vaudrait néant" par "le nom de l'objet supérieur serait une chaîne vide".*

### 10) **Nouveau paragraphe 6.9.1.3**

*Insérer un nouveau paragraphe 6.9.1.3 intitulé "Identificateur demandé" et renuméroter le paragraphe 6.9.1.3 existant ainsi que les paragraphes subséquents.*

#### **6.9.1.3 Identificateur demandé**

Le troisième paramètre est l'identificateur dont l'attribution au nouvel objet est demandée. Cette chaîne deviendra le champ *ID* du rattachement de nom CORBA créé dans le service de dénomination CORBA pour le nouvel objet. Cette chaîne désignera donc, avec le champ *kind* contenu dans le module de rattachement de nom, les noms du nouvel objet par rapport à l'objet supérieur. Si le client fournit une chaîne vide pour ce paramètre, la fabrique choisira automatiquement une valeur d'identification. La fabrique déclenchera une exception de type *duplicateName CreateError* si l'identificateur fourni fait double usage. (En d'autres termes, les deux champs *ID* et *kind* correspondent à un objet déjà contenu dans l'objet supérieur.)

### 11) **Paragraphe 6.9.1.4**

*Remplacer le contenu du paragraphe 6.9.1.4 (intitulé "Nom," et renuméroté par le point précédent) par ce qui suit:*

Le quatrième paramètre doit être utilisé par la fabrique pour renvoyer au client le nom complet de l'objet géré qui vient d'être créé.

### 12) **Paragraphe 9.1**

*Après le 8<sup>e</sup> point, ajouter à la liste le 9<sup>e</sup> point suivant:*

9) Respecter les conventions relatives aux chaînes de caractères d'un paquetage conditionnel, spécifiées au § 6.6.

*Renommer les points suivants.*

### 13) **Annexe A**

*Dans le langage IDL, supprimer les lignes suivantes:*

```
/** A SuspectObject identifies an object that may be the cause of a
failure. It is usually a component of a SuspectObjectList.
@member objectClass          Object class of the suspect object
@member suspectObjectInstance Object instance of the suspect
object
@member failureProbability   Optional failure responsibility
probability from 1 to 100
*/
```

```

struct SuspectObjectType {
    ObjectClassType          objectClass;
    ManagedObject            suspectObjectInstance;
    UnsignedShortTypeOpt    failureProbability;
};

```

*et les remplacer par les lignes suivantes:*

```

/** A SuspectObject identifies an object that may be the cause of a
failure. It is usually a component of a SuspectObjectList.
@member objectClass      Object class of the suspect object
@member suspectObject    Name of the suspect object
@member failureProbability Optional failure responsibility
                          probability from 1 to 100
*/

struct SuspectObjectType {
    ObjectClassType          objectClass;
    MONameType              suspectObject;
    UnsignedShortTypeOpt    failureProbability;
};

```

*Supprimer les lignes suivantes:*

```

valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};

```

*et les remplacer par les lignes suivantes:*

```

valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};

```

*Supprimer les lignes suivantes:*

```

valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};

```

*et les remplacer par les lignes suivantes:*

```

valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};

```

*Supprimer les lignes suivantes:*

```

/** Forward declaration. CORBA uses object references
of type "object" to identify objects. These are used instead of ASN.1
object instances. For network management interfaces, all objects will
inherit from the "ManagedObject" interface. */

interface ManagedObject;

/** MO Set is a set of ManagedObject references. */

typedef sequence <ManagedObject> MOSetType;

/** MO Seq is a sequence of ManagedObject references. */

typedef sequence <ManagedObject> MOSeqType;

```

*et les remplacer par les lignes suivantes:*

```

/** A sequence of names is defined as just a sequence of names. */

```

```
typedef sequence <NameType> NameSeqType;
```

*Après ces lignes:*

```
/** A set of names is defined as a sequence of names. */
```

```
typedef sequence <NameType> NameSetType;
```

*insérer les lignes suivantes:*

```
/** A managed object name is just a name */
```

```
typedef NameType MOnameType;
```

```
/** A sequence of managed object names is defined as just a sequence  
of names. */
```

```
typedef NameSeqType MOnameSeqType;
```

```
/** A set of managed object names is defined as just a set  
of names. */
```

```
typedef NameSetType MOnameSetType;
```

#### **14) Nouvelle Annexe C**

*Ajouter la nouvelle annexe suivante, qui définit les types communs.*

### ANNEXE C

#### **Module IDL des types communs en architecture Corba**

```
#ifndef _itut_x780ct_idl_  
#define _itut_x780ct_idl_  
  
#include <itut_x780.idl>  
#include <itut_x780_1.idl>  
  
#pragma prefix "itu.int"  
  
/**  
This IDL code is intended to be stored in a file named "itut_x780ct.idl"  
located in the search path used by IDL compilers on your system.  
*/  
  
/**  
This module, itut_x780ct, contains common data type for ITU-T CORBA  
framework based information model. This module can be extended by adding  
new data type.  
*/  
module itut_x780ct  
{  
  
// Imports  
  
/**  
Types imported from itut_x780  
*/  
    typedef itut_x780::GeneralizedTimeType GeneralizedTimeType;  
    typedef itut_x780::PerceivedSeverityType PerceivedSeverityType;
```

## // Data Types (X.721)

```
struct CounterThresholdType
{
    long          comparisonLevel;
    long          offsetValue;
    boolean       notificationOnOff;
};

struct NotifyThresholdType
{
    float         threshold;
    boolean       notifyOnOff;
};

struct GaugeThresholdType
{
    NotifyThresholdType    notifyLow;
    NotifyThresholdType    notifyHigh;
};

typedef sequence<GaugeThresholdType> GaugeThresholdSetType;

enum TideMarkChoice
{
    tideMarkChoiceMax,
    tideMarkChoiceMin
};

union TideMarkType switch (TideMarkChoice)
{
    case tideMarkChoiceMax:
        float          maxTideMark;
    case tideMarkChoiceMin:
        float          minTideMark;
};

struct TideMarkInfoType
{
    TideMarkType        currentTideMark;
    TideMarkType        previousTideMark;
    GeneralizedTimeType resetTime;
};
```

## // Data Types (X.739)

```
/**
PerceivedSeverityTypeOpt is an optional type. If the discriminator is true
the value is present, otherwise the value is null.
*/
union PerceivedSeverityTypeOpt switch (boolean)
{
    case TRUE:
        PerceivedSeverityType    val;
};

/**
The SeverityIndicatingThresholdType contains the threshold level, which is to
be applied to counter/gauge attribute. It shall be initialized when the
managed object in which it is included is created and may be modified. An
optional parameter is used to associate the threshold level to the severity
parameter of the emitted notification. The generation of the notification can
be switched off using the boolean parameter notifyOnOff. The severity parameter
is mandatory if notifyOnOff is true.

If a notify-high or notify-low switch is on (true) and the severity indication
value is not present, then the indeterminate value is sent in the notification
unless the managed object behaviour specifies another value.
*/
struct SeverityIndicatingThresholdType
{
```

```

        float                threshold;
        boolean              notifyOnOff;
        PerceivedSeverityTypeOpt severityIndication;
    };

/**
The attribute of SeverityIndicatingGaugeThresholdType has similar behaviour to
the gauge-threshold attribute defined in X.721. The syntax has an added
parameter for indicating the associated severity, as defined in
SeverityIndicatingThresholdType, to the notification triggered by the crossing
of the corresponding threshold level. As an enhancement to the syntax of the
gauge-threshold attribute type it adds an optional severity indication
parameter to the syntax of both the notify-high and notify-low sub-members
within each threshold level member. This attribute type has additional
behaviour associated with these optional perceived severity indication
parameters, which is defined as follows:

- If the notify-high's switch is on (true), the notify-high's severity
indication value shall be reported in the perceived severity parameter of a
notification triggered by the gauge value crossing the notify-high's
gauge-threshold value in the positive going direction.

- If the notify-low's switch is on (true), the notify-low's severity
indication value shall be reported in the perceived severity parameter of a
notification triggered by the gauge value crossing the notify-low's
gauge-threshold value in the negative going direction.

- If both switches are on (true) for a single threshold level, one of the
severity indication values shall be "clear". The severity indicating
gauge-threshold shall only emit a clear event notification if the
corresponding threshold level (either notify-high or notify-low) notification
has been emitted and no other clear notification for this threshold level pair
has been emitted since the previous corresponding threshold level notification
has been emitted.
*/
    struct SeverityIndicatingGaugeThresholdType
    {
        SeverityIndicatingThresholdType notifyLow;
        SeverityIndicatingThresholdType notifyHigh;
    };

/**
The order is insignificant.
*/
    typedef sequence<SeverityIndicatingGaugeThresholdType>
        SeverityIndicatingGaugeThresholdSetType;

    struct TimeIntervalType
    {
        unsigned short day;
        unsigned short hour;
        unsigned short minute;
        unsigned short second;
        unsigned short ms;
    };

/**
The following definitions are translated from X.739 ASN.1 definitions.
*/
    enum TimePeriodChoice
    {
        timePeriodChoiceDays,
        timePeriodChoiceHours,
        timePeriodChoiceMinutes,
        timePeriodChoiceSeconds,
        timePeriodChoiceMilliSeconds,
        timePeriodChoiceMicroSeconds,
        timePeriodChoiceNanoSeconds,
        timePeriodChoicePicoSeconds
    };

```

```
union TimePeriodType switch (TimePeriodChoice)
{
    case timePeriodChoiceDays:
        long day;
    case timePeriodChoiceHours:
        long hour;
    case timePeriodChoiceMinutes:
        long minute;
    case timePeriodChoiceSeconds:
        long second;
    case timePeriodChoiceMilliseconds:
        long ms;
    case timePeriodChoiceMicroSeconds:
        long us;
    case timePeriodChoiceNanoSeconds:
        long ns;
    case timePeriodChoicePicoSeconds:
        long ps;
};

}; // module itut_x780ct

#endif // _itut_x780ct_idl_
```



## 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
Série Q	Commutation et signalisation
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
<b>Série X</b>	<b>Réseaux de données et communication entre systèmes ouverts</b>
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication