



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

T.611

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(11/94)

TERMINAUX POUR LES SERVICES TÉLÉMATIQUES

**INTERFACE DE COMMUNICATION
PROGRAMMABLE APPLI/COM POUR LES
SERVICES DE TÉLÉCOPIE DU GROUPE 3,
DE TÉLÉCOPIE DU GROUPE 4, TÉLÉTEX,
TÉLEX DE MESSAGERIE ÉLECTRONIQUE
ET DE TRANSFERT DE FICHIERS**

Recommandation UIT-T T.611

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (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.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études 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 n° 1 de la CMNT (Helsinki, 1^{er}-12 mars 1993).

La Recommandation révisée UIT-T T.611, que l'on doit à la Commission d'études 8 (1993-1996) de l'UIT-T, a été approuvée le 11 novembre 1994 selon la procédure définie dans la Résolution n° 1 de la CMNT.

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 de télécommunications.

© UIT 1995

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

	<i>Page</i>
PARTIE I – DESCRIPTION GÉNÉRALE.....	1
1 Champ d'application.....	1
2 Définitions et références.....	1
2.1 Définitions de l'interface.....	1
2.2 Définition de fichiers.....	2
2.3 Références.....	3
2.4 Abréviations et acronymes.....	4
2.5 Systèmes d'exploitation.....	6
2.6 Marques déposées.....	6
3 Structure de la présente Recommandation.....	6
3.1 Extension de la Recommandation.....	7
4 Principes généraux.....	7
4.1 Modèle.....	7
4.2 Echange d'informations.....	8
4.3 Environnement de configuration d'interface (ICE).....	10
4.4 Principe de soumission.....	11
5 Caractéristiques de fonctionnement.....	11
5.1 Classes fonctionnelles et profils de service.....	11
5.2 Traitement des erreurs.....	12
5.3 Applications locales multiples et applications de communication multiples.....	13
5.4 Moyens d'identification.....	13
5.5 Service de distribution des fichiers reçus (DRF).....	15
5.6 Commande d'appel – Enregistrement de CA.....	15
6 Descriptions de données relatives aux tâches (TDD).....	23
6.1 Présentation générique des descriptions TDD.....	23
6.2 Description des éléments TDD.....	25
6.3 Identificateur de code (Code-ID).....	25
6.4 Codage en mode texte.....	25
6.5 Traitement de documents.....	35
6.6 Fonctionnalité des descriptions TDD.....	36
7 Méthode d'échange.....	55
7.1 Vue d'ensemble des fonctions de la méthode d'échange de base.....	55
7.2 Fonctions de la méthode d'échange de base.....	58
7.3 Mise en oeuvre des fonctions de la méthode d'échange de base.....	68
8 Formats de transfert.....	70
8.1 Formats de transfert APPLI/COM: ASCII étendu ou normalisé.....	72
8.2 Format de transfert APPLI/COM: T.61.....	73
8.3 Format de transfert APPLI/COM: TIFF.....	73
8.4 Contraintes de service s'appliquant aux formats de transfert.....	82
9 ICE (environnement de configuration d'interface).....	83
9.1 Présentation de l'ICE.....	85
9.2 Obtention de l'accès à l'ICE.....	85
9.3 ICE maître.....	85
9.4 Descripteur de CA.....	86
9.5 Composantes du descripteur de CA.....	86

	<i>Page</i>	
10	Classes et profils fonctionnels.....	88
10.1	Classe fonctionnelle A.....	88
10.2	Classe fonctionnelle B.....	88
10.3	Fonctions supplémentaires.....	90
10.4	Profils de services.....	90
PARTIE II – ENTITÉS DÉPENDANTES DU SERVICE		91
11	Service: Télécopie du groupe 3.....	91
11.1	Éléments syntaxiques propres au service.....	91
11.2	Codage de type texte.....	91
11.3	Fonctionnalités additionnelles	97
11.4	Valeurs du descripteur de l'application de communication (CA-Descriptor)	100
12	Service: Télécopie du groupe 4.....	102
12.1	Éléments syntaxiques propres au service.....	102
12.2	Codage de type texte.....	103
12.3	Fonctionnalités additionnelles	107
12.4	Valeurs du descripteur de l'application de communication (CA-Descriptor)	107
13	Service: Télétex.....	107
13.1	Éléments syntaxiques propres au service.....	107
13.2	Codage de type texte.....	112
13.3	Fonctionnalités additionnelles	115
13.4	Valeurs du descripteur de l'application de communication (CA-Descriptor)	118
14	Service: Télex par télétex.....	119
14.1	Éléments syntaxiques propres au service.....	119
14.2	Codage de type texte.....	120
14.3	Fonctionnalités additionnelles	122
14.4	Valeurs du descripteur de l'application de communication (CA-Descriptor)	123
15	Service: Télex.....	125
15.1	Éléments syntaxiques propres au service.....	125
15.2	Codage de type texte.....	125
15.3	Fonctionnalités additionnelles	127
15.4	Valeurs du descripteur de l'application de communication (CA-Descriptor)	130
16	Service: Courrier électronique	131
16.1	Éléments syntaxiques propres au service.....	132
16.2	Codage de type texte.....	133
16.3	Messagerie de personne à personne.....	142
16.4	Notification de personne à personne.....	153
16.5	Rapport de remise	154
16.6	Correspondance avec éléments de service des systèmes de messagerie.....	156
16.7	Profils du courrier électronique	158
16.8	Valeurs du descripteur de l'application de communication (CA-Descriptor)	159
17	Service: Transfert de fichiers	159
17.1	Éléments syntaxiques propres au service.....	163
17.2	Codage de type texte.....	164
17.3	Fonctionnalités additionnelles	167
17.4	Valeurs du descripteur de l'application de communication (CA-Descriptor)	169

	<i>Page</i>
PARTIE III – BINARY ENCODING SCHEME (SYSTÈME DE CODAGE BINAIRE)	170
18 Description générique en langage C.....	170
18.1 Codage binaire des description TDD.....	170
PARTIE IV – PLATFORM DEPENDENCIES (DÉPENDANCE DES PLATES-FORMES)	194
19 Facteurs de dépendance à l'égard de la mise en oeuvre.....	194
19.1 Mise en correspondance des types de données des descriptions TDD à codage binaire	194
19.2 Méthode d'échange par défaut	194
19.3 Mise en œuvre de la méthode d'échange de primitives	195
Annexe A – Syntaxe pour la présentation et le codage	204
A.1 Syntaxe à forme de Backus-Naur (BNF).....	204
A.2 Notation en langage C	205
Annexe B – Emplacement de l'environnement de configuration d'interface (ICE).....	205
Annex C – Liste des codes d'erreur APPLI/COM	207
Annexe D – Exemples d'échanges de description TDD	209
D.1 Exemple de session d'émission.....	209
D.2 Exemple de session de réception	210
D.3 Exemple de session de suivi (Trace).....	212
Annex E – Exemple d'environnement de configuration d'interface (ICE)	214
Annexe F – Méthode d'échange dans la version 1992.....	216
F.1 Fonctions de la méthode d'échange selon la version 1992	216
Annexe G – Informations spécifiques des services	222
G.1 Ligne d'identification d'appel (CIL) des services de télécopie G4 et de télétext	222
G.2 Identificateur de terminal.....	222
Annexe H – Relevé des formats de transfert et d'émission.....	223
H.1 Formats de transfert en fonction des formats d'émission.....	223
H.2 Formats d'émission associés à un service	223

RÉSUMÉ

La présente Recommandation, telle qu'elle a été approuvée en 1992, définit une interface de communication programmable appelée «APPLI/COM», assurant un accès unifié aux services de télécopie du groupe 3, de télécopie du groupe 4, de télex et de télétexte.

Les auteurs de la révision de la présente Recommandation cherchent à en rendre le texte plus clair et à y inclure les services de messagerie électronique en général, les systèmes de messagerie (MHS), tels qu'ils sont décrits dans la série des Recommandations X.400 de l'UIT-T, en particulier et les services de transfert de fichiers.

Une disposition spéciale a été prise afin d'assurer la compatibilité amont avec la version de 1992 de la présente Recommandation.

**INTERFACE DE COMMUNICATION PROGRAMMABLE APPLI/COM
POUR LES SERVICES DE TÉLÉCOPIE DU GROUPE 3,
DE TÉLÉCOPIE DU GROUPE 4, TÉLÉTEX, TÉLEX DE MESSAGERIE
ÉLECTRONIQUE ET DE TRANSFERT DE FICHIERS**

(révisée en 1994)

PARTIE I – DESCRIPTION GÉNÉRALE

1 Champ d'application

La présente Recommandation définit l'interface de communication programmable (PCI) appelée «APPLI/COM».

Les concepts généraux des interfaces PCI sont définis dans la Recommandation UIT-T F.581 (1992).

La PCI fournit un accès unifié à différents services de télécommunication tels que les services de télécopie du groupe 3 ou d'autres services de télématique.

Actuellement, la présente Recommandation spécifie l'accès aux services de télématique suivants:

- télécopie du groupe 3;
- télécopie du groupe 4;
- télétext;
- téléx¹⁾;
- services de courrier électronique;
- services de transfert de fichiers.

Le principe de cette interface consiste à transférer des messages entre deux entités (à savoir l'application locale et l'application de communication). La présente Recommandation décrit la structure et le contenu de ces messages ainsi que le procédé d'échange. Deux systèmes de codage de message sont définis, l'un, en mode texte, l'autre, en mode binaire.

Il n'est pas obligatoire de disposer de l'interface APPLI/COM pour accéder à un service télématique. Cependant, l'utilisation de cette interface garantit la compatibilité binaire entre les deux entités susmentionnées. En conséquence, les utilisateurs finals bénéficieront de la compatibilité d'interconnexion entre des produits provenant de différents fabricants.

La présente Recommandation fournit un cadre de travail pour les extensions futures, ce cadre conservant la cohérence des fonctionnalités et leur compatibilité amont.

En résumé, la présente Recommandation constitue une API (interface de programmation d'application) de haut niveau qui occulte toutes les particularités de télécommunication, mais donne aux concepteurs d'applications un puissant moyen de commande et de surveillance des activités de télécommunication.

2 Définitions et références

2.1 Définitions de l'interface

Les définitions suivantes s'appliquent à l'interface.

2.1.1 application de communication (CA) (*communication application*): Application de communication fournie par un matériel et/ou un logiciel, permettant d'accéder à des services de télécommunication normalisés. Les services de télécommunication auxquels la présente Recommandation spécifie l'accès sont énumérés dans l'article 1.

¹⁾ L'accès au service téléx n'inclut pas la possibilité de dialogue.

2.1.2 application locale (LA) (*local application*): Une application locale est une application capable de créer des fichiers ou des documents et de gérer éventuellement des dialogues de communication. Les applications locales peuvent être des traitements de texte, des tableurs, des éditeurs graphiques, des éditeurs de fichiers, etc. L'application locale doit émettre des messages conformes à la présente Recommandation.

2.1.3 description de données relatives aux tâches (TDD) (*task data description*): Les descriptions de données relatives aux tâches décrivent la structure des messages échangés entre une application locale et une application de communication (et non la méthode pratique d'échange de ces messages). Une description TDD de demande correspond à un transfert lancé par une application locale vers une application de communication. Une description TDD de réponse correspond à un transfert lancé par une application de communication vers une application locale.

2.1.4 méthode d'échange (EM) (*exchange method*): La méthode d'échange décrit comment les descriptions de données relatives aux tâches sont échangées entre une application de communication et une application locale. La présente Recommandation définit une méthode d'échange générique qui doit être adaptée au système d'exploitation cible par les moyens décrits dans la présente Recommandation. Les environnements de réseaux locaux et de réseaux étendus sont gérés par la méthode d'échange générique.

2.1.5 descripteur d'application de communication: Ensemble d'informations concernant l'utilisation de l'application de communication. Le descripteur de CA décrit les fonctions et les caractéristiques d'une application de communication donnée afin que toute application locale qui utilise cette application de communication sache comment procéder. Le descripteur de CA est un élément de l'environnement de configuration d'interface.

2.1.6 environnement de configuration d'interface (ICE) (*interface configuration environment*): Environnement de configuration d'interface composé de l'environnement ICE de référence et des descripteurs d'application de communication qui lui sont associés. L'environnement ICE de référence énumère toutes les applications de communication qui peuvent être jointes à partir d'une application locale donnée. Le but de l'environnement ICE est d'aider l'application locale à choisir une application de communication adaptée à ses besoins.

2.2 Définition de fichiers

La présente Recommandation traite des divers types de fichiers, à savoir ceux qui sont transférés entre une application locale et une application de communication et ceux qui sont échangés par l'intermédiaire d'un réseau de télécommunication. Les définitions suivantes sont utilisées dans le texte de la présente Recommandation.

2.2.1 fichiers de transfert: Comme le montre la Figure 1, les fichiers de transfert sont les fichiers échangés entre une application locale et une application de communication. Le format de ces fichiers (format de transfert) est défini plus en détail dans la présente Recommandation.

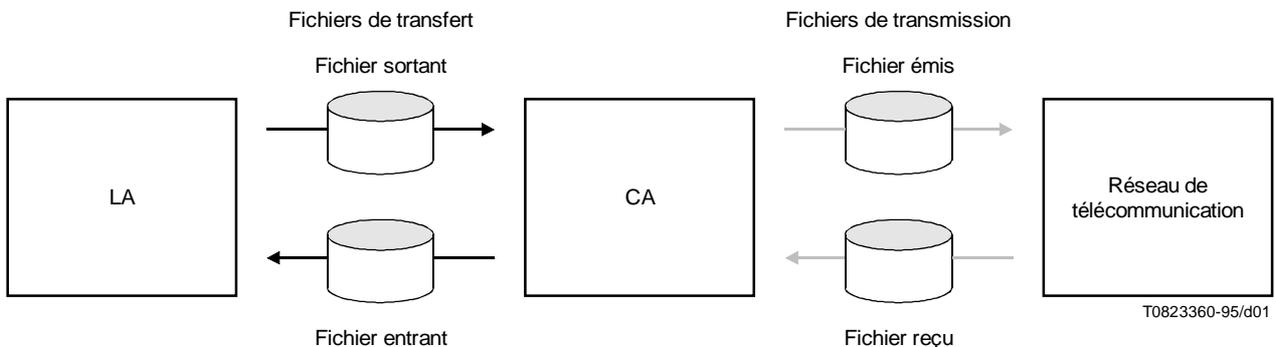


FIGURE 1/T.611

Les divers types de fichiers définis par l'interface APPLI/COM

2.2.1.1 fichier sortant: Fichier que l'application locale transmet à l'application de communication afin que celle-ci le retransmette par un réseau de télécommunication. Le format de ce fichier est l'un des formats de transfert possibles.

2.2.1.2 fichier entrant: Fichier transmis à l'application locale par l'application de communication. En général, il correspond aux fichiers reçus du réseau. Le format de ce fichier est l'un des formats de transfert possibles.

2.2.1.3 format de transfert: Le format de transfert définit la structure des fichiers de transfert. Selon les services de télécommunication utilisés, certains formats de transfert conviennent mieux que d'autres. Les formats de transfert possibles sont définis à l'article 8. Un format de transfert est identifié par le paramètre Convert-ID. L'identification et la sélection d'un format de transfert sont décrites plus en détail au 5.4.5.

2.2.2 fichiers de transmission: Fichiers échangés par une application de communication à travers le réseau, comme le montre la Figure 1. Le format de ces fichiers, le format de transmission, est intrinsèquement défini par le service de télécommunication utilisé.

2.2.2.1 fichier émis: Fichier émis par l'application de communication à travers un réseau de télécommunication dans un format adapté à l'échange, selon le protocole utilisé dans le service de télécommunication.

2.2.2.2 fichier reçu: Fichier créé par l'application de communication à partir d'informations reçues du réseau de télécommunication dans le format employé pour l'échange selon le protocole utilisé dans le service de télécommunication.

2.2.2.3 format de transmission: Le format de transmission définit la structure des fichiers de transmission, en fonction des services de télécommunication utilisés. Le format de transmission est identifié par l'identificateur Type-ID (voir 5.4.6).

2.3 Références

La présente Recommandation comporte des références aux Recommandations UIT-T suivantes:

- Recommandation F.59, *Caractéristiques générales du service télex international.*
- Recommandation F.60, *Dispositions relatives à l'exploitation du service télex international.*
- Recommandation F.160, *Dispositions générales relatives à l'exploitation des services publics internationaux de télécopie.*
- Recommandation F.184, *Dispositions relatives à l'exploitation du service public international de télécopie entre postes d'abonnés avec télécopieurs du groupe 4 (téléfax 4).*
- Recommandation F.200, *Service télétext.*
- Recommandation F.581, *Directives pour la définition des interfaces de communication de programmation: Recommandation de service.*
- Recommandation S.1, *Alphabet télégraphique international n° 2.*
- Recommandation T.4, *Normalisation des télécopieurs du groupe 3 pour la transmission de documents.*
- Recommandation T.6, *Schémas de codage et fonctions de commande de codage de la télécopie pour les télécopieurs du groupe 4.*
- Recommandation T.30, *Procédure pour la transmission de documents par télécopie sur le réseau téléphonique public commuté.*
- Recommandation T.35, *Procédure d'attribution des codes définis par le CCITT.*
- Recommandation T.50, *Alphabet international de référence (ancien Alphabet international n°5 ou A15). Technologie de l'information. Jeux de caractères codés à 7 bits pour l'échange d'informations dans le cas de moyens non normalisés.*
- Recommandation T.51, *Jeux de caractères latins codés pour services de télématique.*
- Recommandation T.52, *Jeux de caractères codés non latins pour les services de télématique.*
- Recommandation T.61, *Répertoire de caractères et jeux de caractères codés pour le service international télétext.*

NOTE – La Recommandation T.61 a été remplacée par les Recommandations de la série T.50 depuis 1993.

- Recommandation T.62, *Procédures de commande pour le service télétext et le service de télécopie du groupe 4.*
- Recommandation T.434, *Format de transfert de fichiers binaires pour les services de télématique.*
- Recommandation T.500, *Présentation générale de la série des Recommandations T.500.*
- Recommandation T.563, *Caractéristiques des télécopieurs du groupe 4.*
- Recommandation T.571, *Caractéristiques des terminaux pour le transfert de fichiers de télématique dans le service télétext.*
- Recommandation X.208, *Spécification de la syntaxe abstraite numéro un (ASN.1).*
- Recommandation X.209, *Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1).*
- Recommandation X.400/F.400: *Services de messagerie: système de messagerie et vue d'ensemble du service.*

La présente Recommandation comporte aussi des références aux normes internationales suivantes:

- ISO/CEI 639-2, *Code pour la représentation de noms de langues.*
- ISO/CEI 9735:1990, *Echange électronique de données pour l'administration, le commerce et le transport (EDIFACT) – Règles syntaxiques au niveau de l'application.*

2.4 Abréviations et acronymes

API	Interface de programmation d'application (<i>application programming interface</i>)
APPLI/COM	Interface entre applications locales et applications de communication
ASCII	Code ASCII (<i>American standard code for information interchange</i>)
ASN.1	Notation de syntaxe abstraite numéro 1 (<i>abstract syntax notation one</i>), telle que définie par les Recommandations UIT-T X.208 et X.209
BFT	Transfert de fichiers binaires (<i>binary file transfer</i>), tel que défini par la Recommandation UIT-T T.434
BNF	Forme de Backus-Naur (<i>Backus-Naur form</i>)
BTM	Mode transparent de base (<i>basic transparent mode</i>), tel que défini par la Recommandation UIT-T T.434
CA	Application de communication (<i>communication application</i>)
CIL	Ligne d'identification de communication (<i>call identification line</i>)
CR	Retour chariot (<i>carriage return</i>)
CTL	Document de commande (<i>control document</i>), tel que défini par la Recommandation UIT-T T.62
DCX	Fichier PCX multipage
DLL	Bibliothèque de liens dynamiques (<i>dynamic link library</i>)
DRF	Distribution des fichiers reçus (<i>dispatch received files</i>)
DTM	Mode transparent au document (<i>document transparent mode</i>), tel que défini par la Recommandation UIT-T T.434
E-Mail	Terme général utilisé pour les services de courrier électronique (<i>electronic mail services</i>)
EBCDIC	Code EBCDIC (<i>extended binary coded data interchange code</i>)
EDI	Echange de données électroniques (<i>electronic data interchange</i>), tel que défini par la Recommandation UIT-T T.571
EM	Méthode d'échange (<i>exchange method</i>)
EMail	Identificateur des services de courrier électronique (<i>electronic mail services</i>)

ESC	Echappement (<i>escape</i>)
FC	Classe fonctionnelle (<i>functional class</i>)
FCA	Classe fonctionnelle A (<i>functional class A</i>)
FCB	Classe fonctionnelle B (<i>functional class B</i>)
FF	Page suivante (<i>form feed</i>)
FT	Identificateur des services de transfert de fichiers (<i>file transfer services</i>)
FX3	Identificateur des services de télécopie du groupe 3 (<i>telex group 3</i>)
FX4	Identificateur des services de télécopie du groupe 4 (<i>telex group 4</i>)
ICE	Environnement de configuration d'interface (<i>interface configuration environment</i>)
IPM	Message de personne à personne (<i>interpersonal message</i>) – message échangé entre usagers par l'intermédiaire de l'agent usager (Terminologie de la Recommandation X.400)
IRA	Alphabet international de référence (T.50) (<i>international reference alphabet</i>)
IRV	Version internationale de référence (T.50) (<i>international reference version</i>)
RNIS	Réseau numérique avec intégration des services
LA	Application locale (<i>local application</i>)
RZL	Réseau de zone locale
LF	Changement de ligne (<i>line feed</i>)
MD	Document de supervision (<i>monitor document</i>); type de document défini par la Recommandation UIT-T T.62
MH	Code de Huffman modifié (<i>modified Huffman code</i>); défini dans la Recommandation UIT-T T.4
MHS	Système de messagerie (<i>message handling system</i>) tel que défini par les Recommandations UIT-T de la série X.400
MR	Code de Read modifié (<i>modified Read code</i>); défini par la Recommandation UIT-T T.4
MS	Mémoire de message (<i>message store</i>) (Terminologie de la Recommandation UIT-T X.400)
MTA	Agent de transfert de messages (<i>message transfer agent</i>) (Terminologie de la Recommandation UIT-T X.400)
MTS	Système de transfert de messages (<i>message transfer system</i>) (Terminologie de la Recommandation UIT-T X.400)
O/R-Name (Nom O/R)	Nom d'expéditeur/de destinataire (<i>originator/recipient name</i>) (Terminologie de la (Nom Recommandation UIT-T X.400)
OPD	Document d'opérateur (<i>operator document</i>), type de document défini par la Recommandation UIT-T T.62
P1	Protocole du système de transfert de messages (<i>MTS transfer protocole</i>) définissant l'enveloppe MHS pour l'échange entre agents MTA défini par les Recommandations UIT-T X.411 (1988) et X.419 (1988, définition de la syntaxe abstraite)
P2	Format de messagerie de personne à personne (IPM) (<i>interpersonal messaging format</i>), défini par la Recommandation UIT-T X.420 (1984, ou profil P22 de 1988)
P3	Protocole d'accès MTS (<i>MTS access protocole</i>), défini par la Recommandation UIT-T X.419 (1988)
P7	Protocole d'accès MS (<i>MS access protocole</i>), défini par la Recommandation UIT-T X.413 (1988)
PCI	Interface de communication programmable (<i>programmable communication interface</i>)
PCX	Divers fichiers internationaux graphiques en points utilisés sur logiciel d'ordinateur personnel

P _{edi}	Format d'échange usager à usager EDI pour X.400, défini par les Recommandations F.435 et X.435
RTPC	Réseau téléphonique public commuté
SP	Espace (<i>space</i>)
TDD	Description de données relatives aux tâches (<i>task data description</i>)
TFT	Transfert de fichiers télématiques (<i>telematic file transfer</i>), tel que défini par la Recommandation UIT-T T.571
TIFF	Format de fichier image balisé (<i>tagged image file format</i>)
TLX	Identificateur des services télex (sans possibilité de dialogue)
TTX	Identificateur des services télételex
TX	Identificateur des services télex obtenus par l'intermédiaire des services télételex
UA	Agent d'utilisateur (<i>user agent</i>) (Terminologie de la Recommandation X.400)
WAN	Réseau métropolitain (<i>wide area network</i>)

2.5 Systèmes d'exploitation

Les versions des systèmes d'exploitation mentionnés dans la présente Recommandation sont les suivantes:

MacOS	Toutes versions
MS-DOS	Version 3.1 ou supérieure
UNIX	Toutes versions
Windows	Version 3.0 ou supérieure
OS/2	Version 1.0 ou supérieure

Tout au long de la présente Recommandation, le terme DOS désigne les versions des systèmes d'exploitation compatibles avec le MS-DOS.

2.6 Marques déposées

Les appellations suivantes sont des marques déposées qui appartiennent respectivement aux différentes sociétés indiquées:

MacOS	Apple Computer, Inc.
MS-DOS	Microsoft Corporation
UNIX	AT&T
TIFF	Aldus Corporation
Windows	Microsoft Corporation
OS/2	International Business Machines Corporation

3 Structure de la présente Recommandation

La présente Recommandation traite les points suivants:

- prescriptions relatives à l'interface;
- définitions;
- principes généraux;
- caractéristiques d'application;
- classes d'instances;
- description de données relatives aux tâches (TDD);
- méthode d'échange;
- fichiers entrants/sortants;
- gestion des communications – Enregistrement de CA.

3.1 Extension de la Recommandation

Afin de s'adapter à l'évolution des technologies de télécommunication, la présente Recommandation a été structurée de manière à accepter les nouveaux services de télécommunication, ou les nouvelles options des services de télécommunication existants sans détruire la présentation générale.

En outre, la présente Recommandation fournit des mécanismes d'extension de manière à enrichir:

- la méthode d'échange entre les applications locales et les applications de communication;
- les messages (les descriptions TDD);
- les formats de transfert,

tout en maintenant la compatibilité avec la Recommandation de base.

4 Principes généraux

L'interface est conforme aux principes suivants:

- être indépendante du matériel informatique (par exemple l'interface peut être prise en charge par un matériel ou un logiciel de communication);
- être indépendante des systèmes d'exploitation et des langages de programmation (la méthode d'échange est la seule partie qui dépende des systèmes d'exploitation et des langages de programmation). La présente Recommandation donne des directives pour réaliser la compatibilité sur les systèmes d'exploitation MS-DOS, OS/2, Windows et Unix. Les autres systèmes d'exploitation seront envisagés sur demande;
- la description formelle des messages d'interface repose sur une description de type BNF²⁾. Divers systèmes de codage peuvent servir à représenter les messages d'interface;
- être orientée soumission de tâches;
- prendre en compte la demande d'applications multiples fonctionnant sur le même serveur, de même que les applications RZL/WAN. L'interface peut être utilisée lorsque plusieurs applications locales et/ou plusieurs applications de communication sont impliquées;
- être extensible et flexible.

4.1 Modèle

L'interaction LA-CA utilise le modèle client-serveur montré dans la Figure 2.

Dans ce contexte:

- l'application de communication est considérée comme un serveur fournissant des services de télécommunication à l'application locale;
- l'application locale est considérée comme le client d'une application de communication qui utilise les services de télécommunication fournis par cette application de communication.

En conséquence:

- en tant que serveur, l'application de communication doit être conforme à l'une des deux classes fonctionnelles définies à l'article 10 et aux méthodes d'échange définies dans la présente Recommandation;
- en tant que client, aucune contrainte ne s'applique à l'application locale en ce qui concerne l'interface APPLI/COM.

Il en résulte que le déclenchement de l'échange d'informations entre application locale et application de communication incombe toujours à l'application locale. L'application locale peut choisir de ne pas être perturbée dans son fonctionnement local par d'éventuels événements émanant de l'application de communication.

4.1.1 Rôle de l'application locale

En ce qui concerne l'interface APPLI/COM, l'application locale peut se diviser en deux parties fonctionnelles:

- le logiciel qui crée les fichiers sortants et/ou lit les fichiers entrants;
- le logiciel qui gère la communication.

²⁾ BNF Forme de Backus-Naur.

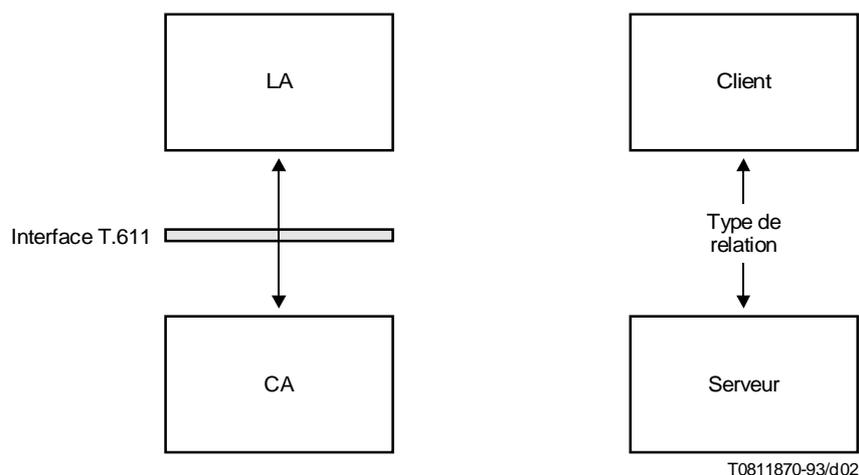


FIGURE 2/T.611

Le modèle de la relation entre application locale et application de communication est de type client-serveur

Ce dernier fournit les fonctions suivantes:

- 1) dialogues homme-machine ou traitement automatique pour l'envoi de fichiers sortants, pour le traitement de fichiers entrants (affichage, impression, sauvegarde), pour le contrôle de l'activité de l'application de communication et pour la demande d'une action particulière de gestion de système et/ou de service;
- 2) conversion de documents en un format de transfert convenant à l'application de communication et réciproquement;
- 3) accès aux caractéristiques facultatives de l'application de communication, comme indiqué dans le descripteur de CA de l'environnement ICE.

NOTE – L'application locale, en tant que client, n'est pas tenue d'utiliser toutes les caractéristiques de l'application de communication.

4.1.2 Rôle de l'application de communication

L'application de communication, en tant que serveur, est chargée des tâches suivantes:

- gestion des communications;
- conversion des formats de fichier du format de transfert en format de transmission (et réciproquement);
- gestion des caractéristiques (éventuelles) de l'application de communication décrites dans le descripteur de CA de l'environnement ICE.

4.2 Echange d'informations

Les informations accessibles à l'interface comprennent:

- *fichiers sortants*: fichiers ou documents remis par l'application locale à l'application de communication pour être transmis par l'application de communication sur un réseau;
- *fichiers entrants*: fichiers ou documents remis par l'application de communication à l'application locale après leur réception par le réseau;
- *enregistrements de CA*: groupes d'informations qui enregistrent tout événement d'émission et de réception géré par l'application de communication.

Les échanges d'informations entre des applications de communication et des applications locales sont effectués par des descriptions TDD de demande et des descriptions TDD de réponse qui appartiennent à l'une des catégories exposées dans le Tableau 1. Ce tableau énumère tous les groupes de descriptions TDD définis dans la présente Recommandation. En particulier, il spécifie si une description TDD de réponse est créée par l'application de communication pour une description TDD de demande donnée. Les descriptions TDD proprement dites sont décrites en détail à l'article 6.

TABLEAU 1/T.611

Groupes de descriptions TDD

Groupe TDD	Réponse?	Commentaire
Envoi (send)	Oui si requis par la demande	Demande à l'application de communication d'envoyer à un ensemble de destinataires un fichier ou plusieurs fichiers à travers le réseau, en utilisant un seul service de télécommunication.
Réception (receive)	Oui	Demande à l'application de communication de récupérer un fichier entrant issu d'un fichier reçu. La description TDD de réponse est remplie avec l'emplacement du fichier entrant et avec le format de transfert.
Suivi (trace)	Oui	Demande à l'application de communication de réaliser une action sur un ensemble d'enregistrements de CA appartenant à tel ou tel état. L'action à prendre est décrite dans la demande suivi.
Soumission (submit)	Oui	Demande à l'application de communication de réaliser une action, comme convertir un fichier à un certain format, ou imprimer un fichier selon les règles propres au service de télécommunication.
Extension (extend)	Oui	Donne la possibilité d'étendre la Recommandation. Ce groupe peut être réalisé sous forme de changements formels à la Recommandation, comme des extensions nationales ou privées.

Le fonctionnement correct de l'interface APPLI/COM exige une description détaillée d'une définition de données et une spécification de l'interaction entre application locale et application de communication. Diverses méthodes d'échange peuvent coexister en fonction des besoins particuliers de l'environnement. On a défini deux méthodes d'échange d'informations (voir l'article 7).

- La première est une méthode d'échange de fichiers facile à mettre en œuvre sur une large gamme d'ordinateurs et de systèmes d'exploitation mais offrant un débit de données inférieur en sortie d'ordinateur.
- La seconde est une méthode d'échange de primitives mettant en œuvre des mécanismes dépendant du système d'exploitation pour transporter les informations. Elle offre un débit de données supérieur mais une portabilité réduite.

Compte tenu de cette situation, on introduit le concept de description de données relatives aux tâches (TDD) (voir la Figure 3). Les descriptions TDD sont une structure de données abstraite, échangée entre application locale et application de communication et décrivant une tâche particulière que l'application de communication doit exécuter ou décrivant une réponse de l'application de communication à l'une de ces tâches.

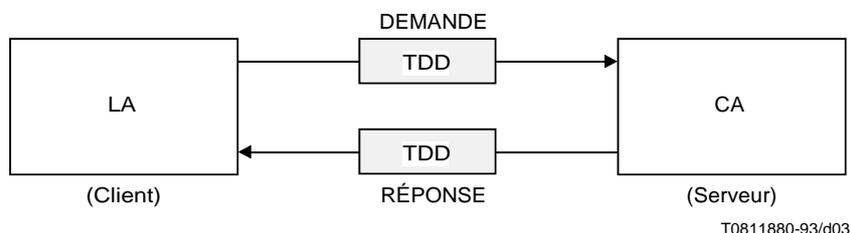


FIGURE 3/T.611

Description de données relatives aux tâches (TDD)

Les descriptions TDD sont indépendantes de la méthode d'échange d'informations utilisée, afin:

- de simplifier les procédures d'essai;
- de permettre de choisir le système de codage des descriptions TDD.

Les descriptions TDD transportent des informations concernant la tâche que l'application locale confie à l'application de communication, ainsi que tous les paramètres appropriés. Comme la communication de ces tâches suit un modèle client-serveur, deux types de descriptions TDD sont effectivement échangés:

- une description TDD de demande, adressée par l'application locale à l'application de communication, décrivant l'action à accomplir;
- une description TDD de réponse, adressée par l'application de communication à l'application locale, décrivant la réponse à une demande antérieure.

De par sa nature, une application locale peut envoyer de nombreuses descriptions TDD de demande sans attendre les descriptions TDD de réponse correspondantes. Certaines demandes ne nécessitent d'ailleurs aucune réponse.

Une application de communication peut gérer les descriptions TDD de demande dans n'importe quel ordre.

4.2.1 Codage des descriptions TDD

Comme indiqué plus en détail au 6.1, les descriptions TDD peuvent être codées sous différentes formes, en mode texte ou à structure binaire.

4.2.2 Format des données échangées

Le format des données échangées entre les applications locales et les applications de communication, c'est-à-dire le format des fichiers sortants et entrants, doit être conforme à des règles de codage et de présentation bien définies. Ces règles sont appelées formats de transfert. Les formats de transfert sont définis en détail à l'article 8.

4.3 Environnement de configuration d'interface (ICE)

L'environnement de configuration d'interface (ICE) est un concept qui est utilisé pour identifier les services offerts par les applications de communication (CA) dans un environnement informatique ainsi que les caractéristiques de chaque application de communication particulière. La structure de l'ICE a deux niveaux:

- 1) L'ICE maître est un fichier universellement accessible, qui donne la liste de toutes les applications de communication disponibles. Il indique la manière d'accéder à l'information de configuration détaillée pour chaque application de communication.
- 2) Le descripteur de CA (*CA-descriptor*) est la source de l'information de configuration détaillée concernant chaque application de communication. L'accès à l'information de descripteur de CA peut se présenter sous forme de fichier ou être créé par des méthodes dynamiques adaptées à l'environnement informatique, telles que des fichiers exécutables, des bibliothèques de liens dynamiques (DDL) ou d'autres approches.

Chaque application de communication possède ses caractéristiques propres, le descripteur de CA fournissant l'accès à une liste complète. La combinaison de l'environnement ICE global et des descripteurs de CA fournit aux applications locales un moyen normalisé d'accéder aux caractéristiques originales et/ou supplémentaires d'application de communication. La Figure 4 représente la relation entre l'environnement ICE maître et les descripteurs de CA.

L'ICE représente une source globale pour toutes les applications locales conformes à la présente Recommandation. L'ICE maître contient les informations d'en-tête suivantes pour chaque application de communication:

- les services pris en charge par l'application de communication;
- la (les) méthode(s) d'échange prise(s) en charge par l'application de communication;
- une ou plusieurs méthodes d'accès pour chaque descripteur de CA.

Le descripteur de CA contient au minimum l'ensemble suivant d'informations de configuration d'interface:

- la méthode d'échange utilisée pour échanger les descriptions TDD entre application locale et application de communication;
- les détails de la méthode d'échange (chemin d'accès, mémoires tampons, points d'entrée);
- le codage des descriptions TDD;
- les classes fonctionnelles;
- les services de télécommunication assurés;
- les services complémentaires offerts par l'application de communication.

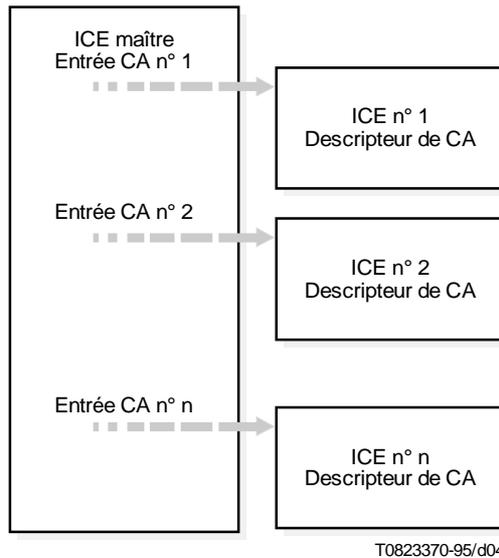


FIGURE 4/T.611
Relation entre ICE maître et descripteur de CA

La mise en œuvre du descripteur de CA est obligatoire pour toute application de communication. N'importe quelle application locale peut utiliser les informations incluses dans l'ICE maître. Les applications locales ne doivent pas modifier les informations contenues dans l'ICE maître ou dans le descripteur de CA.

L'ICE maître est un fichier logique. La syntaxe et le format de l'ICE maître et du descripteur de CA sont décrits à l'article 9. L'emplacement de l'ICE maître dans une plate-forme donnée est indiqué dans l'Annexe B.

4.4 Principe de soumission

Etant donné que les applications de communication et les applications locales peuvent se partager les fonctions requises pour répondre aux normes d'un service de télécommunication, certaines de ces fonctions sont éventuellement redondantes ou manquantes. Citons, à titre d'exemple, les fonctions suivantes:

- vérification de la conformité du format de transfert à certaines prescriptions de service;
- impression d'un document conformément au service de télécommunication par l'intermédiaire duquel il a été (ou sera) transmis;
- conversion de formats de transmission en formats de transfert et réciproquement.

Le principe de soumission permet aux applications locales de profiter des fonctions que l'application de communication assure elle-même, ce qui évite aux applications locales d'assurer des fonctions déjà traitées par l'application de communication. Ainsi, toutes les fonctions requises par les services de télécommunication pour tout équipement conforme seront effectivement assurées.

5 Caractéristiques de fonctionnement

5.1 Classes fonctionnelles et profils de service

Deux classes fonctionnelles (FC) sont définies. Ces classes sont appelées classe fonctionnelle A (FCA) et classe fonctionnelle B (FCB). La classe fonctionnelle B est un strict surensemble de la classe fonctionnelle A. Pour de plus amples précisions, voir l'article 10.

La classe fonctionnelle d'une application de communication doit être explicitement indiquée dans la documentation du fabricant et dans le champ approprié du descripteur de CA.

En outre, l'interface assure l'accès aux caractéristiques supplémentaires de l'application de communication. Ces caractéristiques ne sont pas essentielles en ce qui concerne les Recommandations UIT-T. Chacune de ces caractéristiques supplémentaires peut être utilisée par une application locale, quelle que soit sa classe fonctionnelle, dès l'instant où ces caractéristiques sont offertes par l'application de communication. Les caractéristiques supplémentaires mises en œuvre par une application de communication donnée doivent être explicitement énoncées dans l'environnement ICE (voir l'article 9).

Pour réduire encore les incompatibilités entre différentes applications commercialisées, la présente Recommandation prend en charge le concept de profils de service. Un profil de service, défini pour certains services, regroupe un ensemble de données de caractéristiques de service qui doivent être prises en charge par les instances revendiquant la conformité à la présente Recommandation (voir l'article 10).

5.2 Traitement des erreurs

5.2.1 Erreurs simples

L'application de communication ne doit pas tenir compte des éléments de syntaxe se présentant hors contexte (par exemple, spécification de la vitesse de transmission pour le service télex). Le fonctionnement doit continuer.

5.2.2 Rejet

Toutes les erreurs de syntaxe (par exemple les éléments de syntaxe non reconnus, les éléments de syntaxe manquants, les éléments de syntaxe contradictoires, les occurrences multiples d'un même élément de syntaxe, les paramètres aberrants dans une description TDD), peuvent conduire au rejet de ces descriptions TDD. Un format non valide de fichier sortant peut également conduire au rejet des descriptions TDD correspondantes par l'application de communication. L'application de communication peut aussi détruire le fichier sortant. En outre, l'application de communication peut rejeter les descriptions TDD si les fichiers qui lui sont associés ne sont pas transférés à l'application de communication dans un certain délai.

5.2.3 Domaines de codes d'erreur

Voir le Tableau 2.

TABLEAU 2/T.611

Attribution des codes d'erreur

Code d'erreur	Signification
0000	Succès (pas d'erreur). Doit être pris en charge par toutes les applications de communication
0001-4999	Réservé pour l'usage privé des applications de communication
5000-5999	Erreurs diverses n'entrant dans aucune autre catégorie
6000-6999	Erreurs de syntaxe
7000-7999	Erreurs de ressource et erreurs d'entrée/sortie (système d'exploitation et erreurs provenant du matériel)
8000-8999	Erreurs de conversion et erreurs liées au format de transfert
9000-9999	Erreurs dépendant du service (signaux de service et échecs)

Une liste complète des codes d'erreur attribués dans le cadre de la présente Recommandation est donnée dans l'Annexe C.

5.3 Applications locales multiples et applications de communication multiples

Grâce à l'interface, de multiples applications locales (LA) peuvent être reliées à une ou à plusieurs applications de communication (CA) simultanément. Pour commander l'accès à de multiples applications de communication, il faut définir l'environnement ICE. En fonction de l'ICE, l'application locale suit une procédure d'établissement d'interface en deux étapes lors de la phase d'initialisation:

- l'application locale choisit d'abord une application de communication appropriée en explorant l'environnement de configuration d'interface (ICE);
- ensuite, l'application locale ouvre une session avec l'application de communication choisie.

Une fois que l'application locale a ouvert une session avec l'application de communication, elle est libre d'utiliser n'importe quel service assuré par l'application de communication jusqu'à ce qu'elle ferme la session avec l'application de communication.

5.3.1 Etape 1: choix de l'application de communication

La première étape que l'application locale doit effectuer lors de la procédure de mise en œuvre de l'interface consiste à obtenir l'accès à l'environnement (ICE) qui fournit une liste d'applications de communication accessibles de l'intérieur du système (voir 4.3 et l'article 9).

5.3.2 Etape 2: ouverture de session d'une application locale avec une application de communication

Une fois que l'application locale a choisi une application de communication appropriée à partir de l'environnement ICE, elle doit s'inscrire auprès de l'application de communication choisie. C'est ce qu'on appelle le processus d'ouverture de session. Le processus d'ouverture de session est assuré à l'aide d'un nom d'ouverture de session (login-name) unique et renvoie un identificateur de connexion (connection-ID) calculé par l'application de communication.

Pour ouvrir la session, l'application locale doit s'appuyer sur les informations recueillies dans le descripteur de CA.

Le processus de fermeture de session permet de dissocier l'application locale de l'application de communication: ce processus est appelé par l'application locale lorsque celle-ci souhaite être déconnectée de l'application de communication. A la suite de ce processus, l'identificateur de connexion est détruit par l'application de communication et devient non valide. Aucun accès au nom de cet identificateur ne doit plus être possible pour l'application locale.

Les processus d'ouverture et de fermeture de session facilitent la mise en œuvre de dispositifs de sécurité qui revêtent une importance toute particulière dans les systèmes à usagers multiples. De plus, les processus d'ouverture et de fermeture de session permettent de mettre en œuvre des mécanismes de sécurité entre application locale et application de communication.

5.4 Moyens d'identification

Pour traiter les divers échanges d'informations par l'interface APPLI/COM, il faut identifier sans ambiguïté les entités en communication et les événements de communication. Ces moyens d'identification permettent:

- de distinguer entre les divers événements de télécommunication (COM-ID);
- d'identifier les applications locales en session ouverte avec une application de communication donnée (LA-ID);
- d'identifier les demandes transmises par une application locale à une application de communication donnée (REQ-ID);
- de définir le format de transfert à utiliser (Convert-ID);
- de définir le format de transmission à utiliser (Type-ID).

Ces moyens d'identification sont décrits en détail dans les paragraphes suivants.

5.4.1 Identification des communications d'une application de communication (COM-ID)

Etant donné qu'une application de communication peut traiter différentes demandes émanant de différentes applications locales ou du réseau, il est nécessaire d'identifier chacun de ces événements.

L'identificateur COM-ID est un identificateur unique fourni par l'application de communication et assigné à chaque événement de communication qui s'est produit dans une application de communication. L'identificateur COM-ID est un champ de l'enregistrement de CA (voir 5.6.2).

Un nouvel identificateur COM-ID est créé par l'application de communication lorsqu'un enregistrement de CA est créé, à savoir dans les trois cas ci-après:

- 1) une application locale adresse une demande d'envoi (Send) à l'application de communication;
- 2) une application locale adresse une demande de suivi (Trace), fonction reprogrammation (Reschedule) à l'application de communication;
- 3) l'application de communication traite un fichier reçu.

L'identificateur COM-ID garantit qu'une application locale donnée peut retrouver n'importe quelle demande de transmission programmée, même si le dialogue LA-CA a pris fin pour une raison ou une autre.

5.4.2 Identification des applications locales dans une application de communication (LA-ID)

Comme la présente Recommandation permet à de multiples applications locales ou à de multiples instances d'application locale d'utiliser simultanément une application de communication donnée, les applications locales doivent être identifiées de manière univoque dans une application de communication donnée.

Pour distinguer les applications locales les unes des autres, la présente Recommandation définit l'identificateur LA-ID comme étant l'identificateur unique qui désigne une instance particulière d'application locale communiquant avec une application de communication. L'identificateur LA-ID est un champ de l'enregistrement de CA (voir 5.6.2).

Une application de communication peut refuser de traiter toute description TDD de demande contenant un identificateur LA-ID qui lui est inconnu. Cela permet de contrôler les accès à une application de communication (voir 5.6.1).

L'identificateur LA-ID est affecté de manière statique à l'application locale ou à l'instance d'application locale. La règle d'affectation des identificateurs LA-ID sort du cadre de la présente Recommandation.

5.4.3 Identification des demandes émanant d'applications locales (REQ-ID)

La présente Recommandation permet à une application locale d'adresser des demandes multiples à une application de communication. Etant donné que l'interface permet de remettre des descriptions TDD de réponse à l'application locale dans un ordre différent de celui où les demandes ont été adressées, il est nécessaire d'identifier les descriptions TDD de demande et les descriptions TDD de réponse correspondantes. La présente Recommandation définit donc l'identificateur REQ-ID comme étant l'unique référence de demande assignée à chaque description TDD de demande et à la description TDD de réponse correspondante.

L'identificateur REQ-ID est calculé par l'application locale, à l'aide de tout moyen approprié garantissant l'unicité des identificateurs REQ-ID pour cette application locale. L'algorithme à utiliser pour le calcul des identificateurs REQ-ID sort du cadre de la présente Recommandation. L'identificateur REQ-ID est un champ de l'enregistrement de CA (voir 5.6.2).

Les procédures de restauration qui pourraient être déduites de l'utilisation de l'identificateur REQ-ID sortent du cadre de la présente Recommandation.

5.4.4 Références aux demandes issues de l'application locale (REQ-REF)

La référence à une demande de LA (REQ-ID) est utilisée dans les descriptions TDD suivi (Trace) pour faire référence à de précédentes descriptions TDD d'émission (Send) et/ou de réception (Receive).

5.4.5 Identification des formats de transfert (Convert-ID)

Le format de transfert – qui sert à transmettre un document entre application locale et application de communication – est identifié par l'identificateur Convert-ID. Pour chaque service, la présente Recommandation définit un ensemble obligatoire de divers identificateurs Convert-ID, qui doivent être acceptés par l'instance de l'application de communication. A ce sujet, voir également le Tableau H.1 ainsi que les articles concernant ce service particulier dans la Partie II de la présente Recommandation.

5.4.6 Identification de formats de transmission (Type-ID)

Le format de transmission – celui qui sert à transmettre un document à travers le service – est identifié par l'identificateur Type-ID. Chaque service définit son propre ensemble de formats de transmission. A ce sujet, voir également les articles concernant ce service particulier dans la Partie II de la présente Recommandation.

5.5 Service de distribution des fichiers reçus (DRF)

Certains services de télécommunication n'offrent pas de services complémentaires de sous-adressage. Sur les systèmes où l'application de communication peut être utilisée par plusieurs applications locales simultanément, il est nécessaire d'assister l'acheminement des documents entrants aux destinataires prévus. C'est ce qu'on appelle le processus de distribution des fichiers reçus (DRF).

Lorsqu'une application de communication reçoit un fichier du réseau, ce fichier doit être affecté à une seule application locale destinataire. Seule l'application locale destinataire peut alors accéder au fichier reçu. Le choix de l'application locale destinataire est un processus privé de l'application de communication, qui sort du cadre de la présente Recommandation.

L'identificateur LA-ID de l'application locale destinataire est alors affecté à l'enregistrement de CA correspondant au fichier reçu.

Si l'application de communication offre le service complémentaire de distribution des fichiers reçus (DRF), l'application locale destinataire peut distribuer les fichiers reçus à l'application locale appropriée au moyen de la demande de suivi: DISTRIBUTION (Trace:DISPATCH) (voir 6.6.3). L'identificateur LA-ID de l'application locale à laquelle le fichier reçu est distribué est alors affecté à l'enregistrement de CA.

Cette caractéristique de l'application de communication étant facultative, la prise en charge du service DRF doit être déclarée dans le descripteur de CA (voir l'article 9).

Si l'application de communication n'assure pas le service complémentaire DRF, elle affecte (par tout moyen approprié) l'enregistrement de CA à toute application locale utilisant le champ LA-ID. Lorsque le sous-adressage est assuré par le service de télécommunication, l'application de communication affecte automatiquement l'enregistrement de CA à l'application locale du destinataire prévu.

Les algorithmes à utiliser pour distribuer les appels entrants aux applications locales appropriées sortent du cadre de la présente Recommandation.

5.6 Commande d'appel – Enregistrement de CA

Etant donné que la méthode d'échange entre une application locale et une application de communication est fondée sur un modèle client-serveur, l'application locale prend toujours l'initiative d'adresser des descriptions TDD de demande à une application de communication. Les descriptions TDD de réponse issues de l'application de communication ne sont pas spontanées; l'application locale doit interroger l'application de communication pour savoir si des descriptions TDD de réponse sont disponibles.

La présente Recommandation indique les moyens permettant à une application locale de suivre tout événement de communication qui se produit dans une application de communication; les applications locales ne sont pas tenues d'utiliser ces moyens.

Un événement d'application de communication rassemble des informations telles que la date et l'heure, l'expéditeur, le(s) destinataire(s), le service de communication, etc.

L'enregistrement de CA est un ensemble fonctionnel d'informations stockées par une application de communication pour traiter une demande Send ou un appel entrant en provenance du réseau. Ces informations sont stockées dans des champs, dont chacun a un objectif particulier.

Un enregistrement de CA est créé par l'application de communication lorsque l'un des événements suivants se produit:

- l'application de communication reçoit une demande Send issue d'une application locale; dans ce cas, si la demande Send contient plusieurs destinataires, l'application de communication doit étendre la liste des destinataires et créer le même nombre d'enregistrements de CA que le nombre de destinataires dans la liste;
- l'application de communication reçoit une demande suivi:REPROGRAMMATION (Trace:RESCHEDULE) en provenance d'une application locale;
- l'application de communication reçoit un appel entrant en provenance du réseau.

Eventuellement, une application de communication peut créer un enregistrement de CA dans l'état «échec» si elle rencontre des conditions d'erreur qui ne sont pas directement la conséquence d'une demande d'émission (Send) ou d'un appel entrant.

Les enregistrements de CA sont détruits lorsque l'application locale émet des demandes Trace:PURGE ou par des moyens propres à l'application de communication.

NOTE – Un enregistrement de CA ne peut être détruit qu'une fois parvenu à un état final, c'est-à-dire à l'état «récupéré», «envoyé» ou «échec» (voir aussi les Figures 5 et 6).

L'enregistrement de CA est une structure interne de l'application de communication. Le mode de mise en œuvre de l'enregistrement de CA sort du cadre de la présente Recommandation.

5.6.1 Contrôle d'accès d'enregistrement de CA

Toutes les actions déclenchées par les applications locales sur des enregistrements de CA sont exécutées à l'aide de la demande Trace. Selon sa configuration, une application de communication peut décider de cacher certaines informations à une application locale. Par exemple, une application de communication peut refuser l'accès à des enregistrements de CA «différés» qui proviennent d'une autre application locale.

Pour faciliter le contrôle d'accès à une application de communication, la présente Recommandation prévoit un mécanisme permettant d'identifier les applications locales au moyen de l'identificateur LA-ID. Par exemple, toutes les demandes Trace adressées par les applications locales contiennent un identificateur LA-ID.

Grâce à ce mécanisme, une application de communication donnée peut décider de limiter ou d'étendre l'utilisation de certaines demandes de contrôle d'accès à un seul ensemble particulier d'applications de communication. Par exemple, la possibilité d'utiliser la demande Trace:DISPATCH (suivi:DISTRIBUTION), peut être offerte à une seule application locale ou à un ensemble d'applications locales différentes, selon la configuration du système; l'accès à la demande Trace:PURGE peut être également réservé à une seule application locale pour des raisons administratives.

L'auteur du logiciel d'application de communication doit préciser, dans sa documentation, comment les (éventuels) contrôles d'accès sont traités et, le cas échéant, comment ces contrôles d'accès peuvent être personnalisés pour s'adapter à la configuration des installations d'usager. L'adaptation personnalisée des applications de communication doit être effectuée par des moyens spécifiques qui sortent du cadre de la présente Recommandation.

5.6.2 Champs de l'enregistrement de CA

L'enregistrement de CA contient une liste minimale de champs qui doivent être acceptés par toutes les applications de communication. Les applications de communication peuvent accepter des champs supplémentaires; ces champs doivent être déclarés dans l'environnement ICE.

Le Tableau 3 indique la liste minimale de champs d'enregistrement de CA que toute application de communication doit accepter, dans l'ordre indiqué:

TABLEAU 3/T.611

Liste minimale des champs de l'enregistrement de CA

Nom du champ	Syntaxe	Objet
COM-ID	COM-ID	Contient l'identificateur de communication unique affecté à l'enregistrement CA
LA-ID	La-id	Assigne l'enregistrement CA à la LA qui en est à l'origine ou à laquelle il est destiné
REQ-ID	Req-id	Contient la référence de la demande
STATE	Descripteur d'état	Indique l'état actuel de l'enregistrement CA
DIRECTION	«Emission»/«Réception»	Indique si l'enregistrement CA a été créé pour une transmission ou une réception
NOTE – Les fabricants d'applications de communication peuvent inclure la prise en charge de champs supplémentaires (par exemple contenant des informations de taxation, d'horodatage, des adresses). Voir également 6.6.3.2.		

L'enregistrement de CA offre à une application locale la possibilité de commander l'application de communication avec laquelle elle a ouvert une session. Un enregistrement de CA est à tout moment dans l'un des 6 états suivants (voir le Tableau 4).

TABLEAU 4/T.611

Etats de l'enregistrement de CA

Nom de l'état	Concerne	Signification
«retardé»	Transmission	Le ou les documents associés n'ont pas encore été traités par la CA. Ils attendent que la CA les place dans l'état envoi
«en cours d'envoi»	Transmission	Le ou les documents associés ont été traités par la CA pour transmission; la CA n'a pas encore fini de les traiter (car ils sont en cours d'envoi ou la transmission a échoué et la CA effectuera bientôt d'autres tentatives)
«envoyé»	Transmission	Le ou les documents associés ont bien été envoyés au destinataire
«échec»	Transmission, réception	Le ou les documents associés n'ont pu être transmis (ou) des erreurs se sont produites lors de la réception (ou) une erreur interne s'est produite dans la CA
«réception»	Réception	L'enregistrement décrit un document qui est en cours de réception par la CA ou a été reçu mais n'est pas encore récupéré par la LA destinataire
«récupéré»	Réception	Le document reçu a été récupéré par la LA destinataire au moyen de la demande Receive

Les transitions d'un état à un autre sont régies par des actions internes à l'application de communication, qui émanent du réseau de communication ou qui sont déclenchées par l'application locale.

Les applications locales peuvent lire les champs d'état d'un enregistrement de CA au moyen de la demande Trace:COPY. Une application locale particulière ne peut accéder qu'aux enregistrements qui ont une valeur d'identificateur LA-ID conforme (sauf si l'application de communication étend les droits d'accès de certaines applications locales). Cela garantit qu'une application locale particulière ne pourra consulter que ses propres enregistrements de CA.

La seule exception à cette règle est lorsqu'une application de communication n'utilise pas le service DRF (voir 5.5), auquel cas n'importe quelle application locale peut accéder à n'importe quel enregistrement de CA qui se trouve dans l'état de réception.

Les paragraphes suivants décrivent les transitions d'état d'un enregistrement de CA en liaison avec les événements d'émission et de réception qui se produisent dans l'application de communication.

5.6.3 Processus d'émission – Transitions d'état

Les émissions sont déclenchées par des demandes d'envoi (Send). Lorsque l'application de communication reçoit une demande Send valide d'une application locale en session ouverte avec elle, cette application constitue autant d'enregistrements de CA que de destinataires contenus dans la demande d'envoi.

Les émissions peuvent être également déclenchées par des émissions de «reprogrammation» qui ont précédemment échoué, au moyen d'une demande Trace:RESCHEDULE. Dans certains cas, cela peut éviter le délai nécessaire pour la conversion des documents en formats de transmission appropriés car ces documents ont déjà été convertis lors d'une demande Send antérieure.

L'application de communication doit affecter un COM-ID à chaque nouvel enregistrement de CA. Si une demande Send indique «destinataires multiples», l'application de communication affecte un COM-ID différent à chaque enregistrement de CA résultant (un par destinataire). L'application de communication met également le champ d'état à l'état «retardé». Elle copie ensuite toutes les informations émanant de la demande Send dans l'enregistrement de CA.

L'application de communication explore à son propre rythme la liste des enregistrements de CA à l'état «retardé» et traite l'un d'entre eux. Le choix de l'enregistrement de CA «retardé» à traiter en premier lieu repose sur les dates et les heures programmées qui ont été spécifiées dans le champ <SendTime> de la demande Send (voir le Tableau 13, paragraphe 6.2). L'enregistrement de CA choisi passe à l'état «en cours d'envoi».

L'application de communication transmet, un par un, tous les documents contenus dans l'enregistrement de CA «en cours d'envoi». Si l'émission échoue suite à des erreurs de transmission, l'application de communication peut maintenir l'enregistrement dans l'état «en cours d'envoi» aussi longtemps qu'elle répète les tentatives d'émission.

Si l'émission échoue finalement, l'enregistrement de CA passe à l'état «échec». Si l'émission réussit finalement, l'enregistrement de CA passe à l'état «envoyé». Entre deux tentatives sur le même enregistrement de CA, l'application de communication peut choisir un autre enregistrement de CA à l'état «retardé» et le traiter comme indiqué ci-dessus. En conséquence, plusieurs enregistrements de CA peuvent être dans l'état «en cours d'envoi» à un moment donné.

La Figure 5 illustre ces transitions d'état.

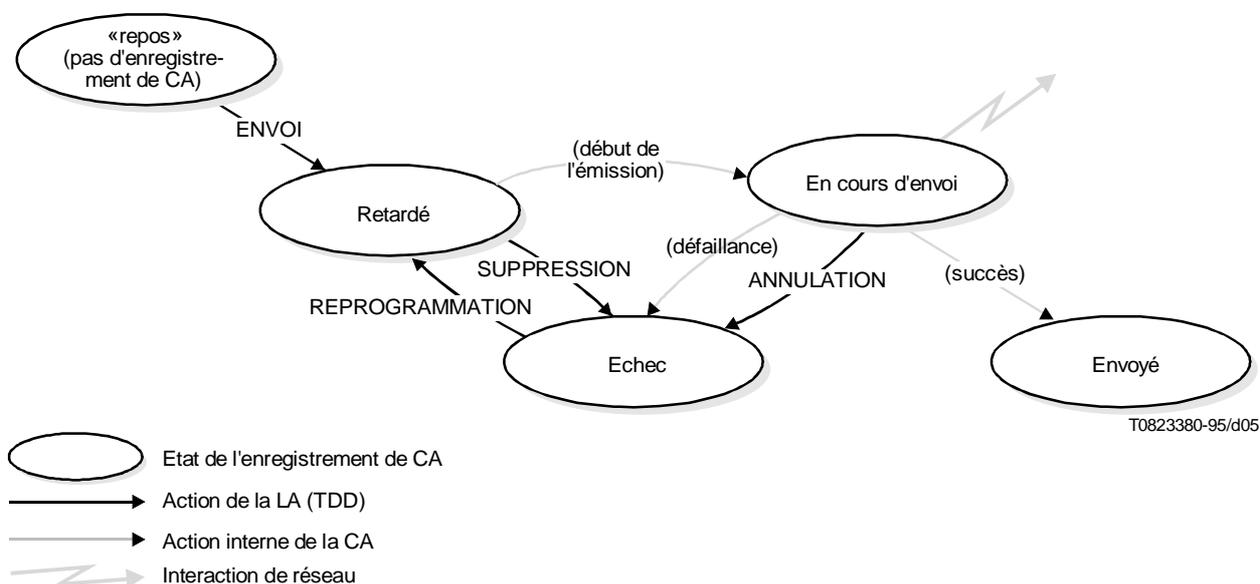


FIGURE 5/T.611
Transitions d'état de l'enregistrement de CA (émissions)

5.6.4 Processus de réception – Transitions d'état

Lorsqu'une application de communication reçoit une communication entrante en provenance du réseau, elle constitue un enregistrement de CA. Un identificateur COM-ID unique est affecté à cet enregistrement de CA et mis à l'état «réception». Toutes les informations concernant l'appel entrant (expéditeur, date et heure, etc.) sont alors copiées dans les champs appropriés de l'enregistrement de CA.

S'il se produit une défaillance en phase de réception, l'enregistrement de CA passe à l'état «échec». Dans le cas contraire, l'enregistrement de CA passe à l'état «récupéré» lorsqu'une application locale l'a récupéré au moyen de la demande de réception (Receive).

La demande de suivi: DISTRIBUTION (Trace: DISPATCH) n'a pas d'effet sur l'état de l'enregistrement de CA, c'est-à-dire que celui-ci reste à l'état «réception»; l'enregistrement de CA n'est plus visible par «l'application locale distributrice» et devient visible par l'application locale destinataire.

La Figure 6 illustre ces transitions d'état.

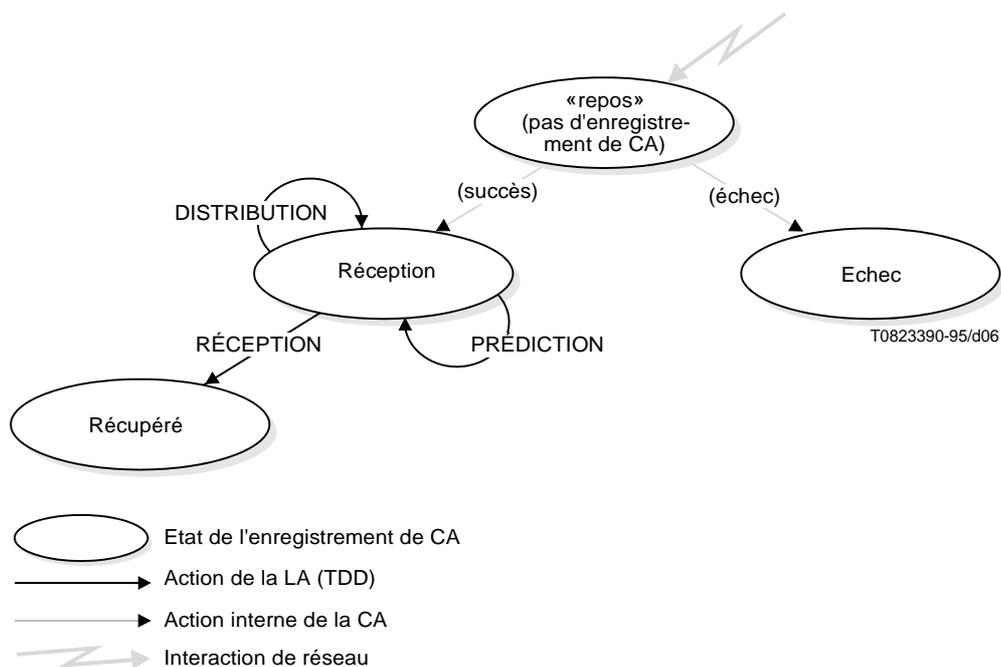


FIGURE 6/T.611
Transitions d'état de l'enregistrement de CA (réception)

5.6.5 Actions – Conventions de notation

Pour des raisons de lisibilité, les conventions suivantes s'appliquent à la notation des diverses actions faisant appel à des enregistrements de CA (voir le Tableau 5).

TABLEAU 5/T.611
Notation des actions faisant appel à l'enregistrement de CA

Notation	Signification
Dem. Trace:DELETE	SUPPRIMER la demande de Trace TDD
LA destinataire	Le destinataire prévu de l'appel entrant. Le destinataire peut être déterminé soit automatiquement (exemple: mécanisme de sous-adressage) ou manuellement. Le traitement automatique appartient à l'application de communication alors que la distribution manuelle peut être commandée par l'interface (voir 10.6.1)
LA expéditrice	La LA qui a établi l'enregistrement CA (au moyen d'une demande Send)
Transmettre	La LA tente d'exécuter l'action «transmettre» sur l'enregistrement CA
NOTE – L'application de communication peut permettre aux applications locales de se comporter en application locale destinataire ou d'origine, ce qui peut être utile dans de nombreuses applications (par exemple, administration).	

5.6.6 Actions – Emissions

Le présent paragraphe décrit toutes les actions qui influent sur les divers états des enregistrements de CA intervenant dans les émissions.

5.6.6.1 Etat retardé

Un enregistrement de CA est dans l'état «retardé» tant qu'il n'a pas été traité par l'application de communication (c'est-à-dire qu'aucune tentative n'a été faite pour l'émettre) ou que l'application locale expéditrice ne l'a pas effacé. Les actions suivantes peuvent s'appliquer à un enregistrement de CA à l'état «retardé» (voir le Tableau 6).

TABLEAU 6/T.611

Actions d'émission pour l'état «retardé»

Action	Expéditeur	Objet	Etat résultant
Dem. Trace:DELETE	LA expéditrice	Supprimer une demande Send. Cette action n'influe que sur un enregistrement CA à la fois	«échec»
Dem. Trace:COPY	LA expéditrice	Créer un fichier logique contenant une copie des enregistrements CA «retardés». L'action s'applique à tous les enregistrements CA «retardés» émanant d'une même LA	«retardé»
Transmettre	CA	La CA décide de traiter l'enregistrement CA car le moment est venu de le faire	«en cours d'envoi»

5.6.6.2 Etat en cours d'envoi

Un enregistrement de CA est dans l'état «en cours d'envoi» lorsque l'application de communication tente d'émettre les informations qu'il contient.

Si la tentative échoue, l'enregistrement de CA peut rester à l'état «en cours d'envoi» en attendant la tentative suivante; l'application de communication devra retarder la tentative suivante, eu égard au «délai de nouvelle tentative», sauf si le «nombre maximal de tentatives» est atteint, auquel cas l'enregistrement de CA est mis à l'état «échec».

Les actions suivantes peuvent s'appliquer à un enregistrement de CA à l'état «en cours d'envoi» (voir le Tableau 7).

TABLEAU 7/T.611

Actions d'émission pour l'état «en cours d'envoi»

Action	Expéditeur	Objet	Etat résultant
Dem. Trace:CANCEL	LA expéditrice	Annuler une demande Send; cela n'influe que sur un enregistrement CA à la fois	«échec»
Dem. Trace:COPY	LA expéditrice	Créer un fichier logique contenant une copie des enregistrements CA «en cours d'envoi»	«en cours d'envoi»

5.6.6.3 Etat envoyé

Un enregistrement de CA est dans l'état «envoyé» lorsque l'application de communication a réussi à transmettre les informations qu'il contient.

Les actions suivantes peuvent s'appliquer à un enregistrement de CA à l'état «envoyé» (voir le Tableau 8).

TABLEAU 8/T.611

Actions d'émission pour l'état «envoyé»

Action	Expéditeur	Objet	Etat résultant
Dem. Trace:PURGE	LA expéditrice	Supprimer tous les enregistrements CA à l'état «envoyé»	Sans objet car les enregistrements de CA ont été supprimés (purgés)
Dem. Trace:COPY	LA expéditrice	Créer un fichier logique contenant une liste des registres CA «envoyés»	«envoyé»

5.6.6.4 Etat échec

Un enregistrement de CA est à l'état d'«échec» lorsque l'application de communication n'a pu, pour une raison ou une autre, émettre les informations qu'il contient.

NOTE – L'état d'échec s'applique également au processus de réception. Voir ci-dessous les informations qui s'y appliquent.

Les actions suivantes peuvent s'appliquer à un enregistrement de CA à l'état d'«échec» (voir le Tableau 9).

TABLEAU 9/T.611

Actions d'émission pour l'état d'«échec»

Action	Expéditeur	Objet	Etat résultant
Dem. Trace:PURGE	LA expéditrice	Supprimer des enregistrements CA dans l'état «échec»	Sans objet car les enregistrements de CA ont été supprimés (purgés)
Dem. Trace:RESCHEDULE	LA expéditrice	Demander de traiter à nouveau une demande ayant échoué; il est possible, à cet égard, de profiter des précédentes conversions de documents	«retardé»
Dem. Trace:COPY	LA expéditrice	Créer un fichier logique contenant une copie de la liste des enregistrements CA ayant «échoué»	«échec»

5.6.7 Actions – Réceptions

Le présent paragraphe décrit toutes les actions qui influent sur les états des enregistrements de CA liés à la réception.

5.6.7.1 Etat de réception

Un enregistrement de CA est dans l'état de «réception» lorsque l'application de communication a reçu avec succès un appel entrant en provenance du réseau et que l'enregistrement de CA n'a pas encore été distribué.

Les actions suivantes peuvent s'appliquer à un enregistrement de CA dans l'état de «réception» (voir le Tableau 10).

TABLEAU 10/T.611

Actions de réception pour l'état «réception»

Action	Expéditeur	Objet	Etat résultant
Dem. RECEIVE	LA destinataire	Récupérer le ou les documents propres à l'enregistrement CA (ainsi que les informations correspondantes dans la réponse TDD)	«récupéré»
Dem. Trace:PREVIEW	N'importe quelle LA	Recevoir une copie du ou des documents propres à l'enregistrement CA. L'enregistrement CA reste dans l'état «réception»	«réception»
Dem. Trace:DISPATCH	LA destinataire ou Admin.	Affecter un ou plusieurs LA-ID à l'enregistrement CA reçu (voir Note ci-dessous)	«réception»
Dem. Trace:COPY	LA destinataire	Créer un fichier logique contenant une copie de la liste des enregistrements de CA dans l'état «réception», c'est-à-dire n'ayant pas encore été récupérés. Une LA destinataire ne «voit» que ses propres enregistrements CA, c'est-à-dire ceux qui correspondent à son LA-ID	«réception»
NOTE – La gestion de l'option d'Administration appartient en propre à l'application de communication.			

5.6.7.2 Etat récupéré

Un enregistrement de CA est dans l'état «récupéré» lorsqu'il a été récupéré par une application locale ayant un identificateur LA-ID correspondant, grâce à une demande Receive.

Les actions suivantes peuvent s'appliquer à un enregistrement de CA dans l'état «récupéré» (voir le Tableau 11).

TABLEAU 11/T.611

Actions de réception pour l'état «récupéré»

Action	Expéditeur	Objet	Etat résultant
Dem. Trace:PURGE	LA destinataire	Supprimer les enregistrements de CA qui se trouvent dans l'état «récupéré»	Sans objet car les enregistrements de CA ont été supprimés (purgés)
Dem. Trace:COPY	LA destinataire	Créer un fichier logique contenant une copie de la liste des enregistrements de CA dans l'état «récupéré». Une application locale destinataire ne «voit» que ses propres enregistrements de CA appropriés, c'est-à-dire ceux qui correspondent à la LA-ID	«récupéré»

5.6.7.3 Etat d'échec

Un enregistrement de CA est dans l'état d'«échec» lorsque l'application de communication n'a pu, pour une raison ou pour une autre, recevoir un appel entrant en provenance du réseau.

NOTE – L'état d'échec s'applique également au processus de transmission. Voir 5.6.6.4 pour les informations correspondantes.

Lorsqu'une application de communication n'a pu recevoir un appel entrant, elle fait passer l'enregistrement de CA de l'état «réception» à l'état «échec».

Lorsqu'un enregistrement de CA est dans l'état «échec», aucun identificateur LA-ID ne lui est affecté. Selon la configuration, l'application de communication peut affecter (mais elle n'est pas tenue de le faire), par tout moyen approprié, un identificateur LA-ID aux enregistrements à l'état «échec». La demande de suivi:REPROGRAMMATION (Trace:RESCHEDULE) ne s'applique pas aux enregistrements de CA qui ont été reçus à l'état «échec».

Le Tableau 12 décrit les actions qui peuvent s'appliquer aux enregistrements de CA à l'état «échec».

TABLEAU 12/T.611

Actions de réception pour l'état d'«échec»

Action	Expéditeur	Objet	Etat résultant
Dem. Trace:PURGE	LA destinataire	Supprimer tous les enregistrements de CA qui se trouvent dans l'état «échec»	Sans objet car les enregistrements de CA ont été supprimés (purgés)
Dem. Trace:COPY	LA destinataire	Créer un fichier logique contenant une copie de la liste des enregistrements de CA dans l'état «échec». Une application locale destinataire ne «voit» que ses propres enregistrements de CA appropriés	«échec»

6 Descriptions de données relatives aux tâches (TDD)

La fonctionnalité offerte à l'interface par l'application de communication est appelée par des descriptions TDD qui sont transmises sous la forme de demandes de l'application locale à l'application de communication et en sens inverse, selon la fonction appelée, sous la forme de réponses.

Le présent article décrit la sémantique, la fonctionnalité, la syntaxe et le codage des diverses descriptions TDD de demande et de réponse.

6.1 Présentation générique des descriptions TDD

Les descriptions TDD peuvent être codées sous différentes formes. Certaines se fondent sur les descriptions textuelles, par exemple ASCII ou EBCDIC. D'autres systèmes de codage ne peuvent pas être représentés sous forme de texte. Parmi ceux-ci, le système de codage binaire, que la Partie III de la présente Recommandation présente en langage C.

Les diverses descriptions TDD seront décrites par la suite de manière générique en syntaxe de type BNF. Cette syntaxe est expliquée de manière détaillée dans A.1.

Pour faciliter la compréhension, les paragraphes qui suivent utilisent le codage en mode texte avec la syntaxe générique de type BNF donnée ci-dessous. Cependant, le système de codage binaire (défini dans la Partie III de la présente Recommandation) peut également être utilisé pour les réalisations.

```

<TDD> :=
    <TDD Header>
    <Send> | <Receive> | <Trace> | <Submit> | <Extend>

<TDD Header> :=
    -- dépend du codage utilisé. Pour le codage en mode texte voir 6.4.3,
    -- pour le système de codage en mode binaire, voir le paragraphe approprié
    -- de la Partie III de la présente Recommandation

<Send> :=
    <SendTDD> | <SendAckTDD>

<Receive> :=
    <ReceiveTDD>

<Trace> :=
    <DeleteTDD> | <CopyTDD> | <CancelTDD> | <PurgeTDD> |
    <RescheduleTDD> | <DispatchTDD> | <PreviewTDD>

```

<Submit> := <PrintTDD> | <ConvertTDD> | <CheckTDD>

<Extend> := <ExtendTDD> | <NationalTDD> | <PrivateTDD>

<SendTDD> := <SendFunction> <LaId> <ReqId> <Service>
 [<SendTime>] [<Comment>] [<LastTime>] [<UserKey>]
 <ServiceDependentKeywordsSend>

<ServiceDependentKeywordsSend> :=
 -- défini pour chaque service dans l'article approprié de la Partie II

<SendAckTDD> := <SendAckFunction> <LaId> <ReqId> <Service> <Error>
 <Status> [<ComId>] [<SendTime>] [<Comment>] [<LastTime>]
 [<UserKey>] [<Minor>] [<Warning>]
 <ServiceDependentKeywordsSendack>

<ServiceDependentKeywordsSendack> :=
 -- défini pour chaque service dans l'article approprié de la Partie II

<ReceiveTDD> := <ReceiveFunction> <LaId> <ReqId> <Error> <Status> [<ComId>]
 [<ReceiveTime>] [<Service>] [<Delete>] [<Minor>] [<Warning>]
 <ServiceDependentKeywordsReceive>

<ServiceDependentKeywordsReceive> :=
 -- défini pour chaque service dans l'article approprié de la Partie II

<DeleteTDD> := <DeleteFunction> <LaId> <ReqId> <Error> (<ComId> | <ReqRef>)
 [<Minor>] [<Warning>]

<CopyTDD> := <CopyFunction> <LaId> <ReqId> <Error> <State> <Target> <Layout>
 (<ComId> | <ReqRef>) [<Minor>] [<Warning>] [<Previewed>]
 [<Dispatched>] [<Direction>] [<Time_Range>] [<Service>]
 <ServiceDependentKeywordsCopy>

<ServiceDependentKeywordsCopy> :=
 -- défini pour chaque service dans l'article approprié de la Partie II

<CancelTDD> := <CancelFunction> <LaId> <ReqId> <Error> (<ComId> | <ReqRef>)
 [<Minor>] [<Warning>]

<PurgeTDD> := <PurgeFunction> <LaId> <ReqId> <Error> <State> (<ComId> | <ReqRef>)
 [<Minor>] [<Warning>]

<RescheduleTDD> := <RescheduleFunction> <LaId> <ReqId> <Error> (<ComId> | <ReqRef>)
 [<Address>] [<SendTime>] [<LastTime>] [<Minor>] [<Warning>]

<DispatchTDD> := <DispatchFunction> <LaId> <NewLa> {<NewLa>} <ReqId> <Error>
 <ComId> [<Minor>] [<Warning>]

<PreviewTDD> := <PreviewFunction> <LaId> <ReqId> <Target> <Convert> <Error> <ComId>
 [<Minor>] [<Warning>]

<PrintTDD> := <PrintFunction> <LaId> <ReqId> <Error> <FileName> <InFormat>
 [<Printer>] [<Minor>] [<Warning>]

<ConvertTDD> := <ConvertFunction> <LaId> <ReqId> <Error> <FileName> <Target>
 <InFormat> <OutFormat> [<Minor>] [<Warning>]

<CheckTDD> := <CheckFunction> <LaId> <ReqId> <Error> <FileName> <Check> [<Minor>]
 [<Warning>]

```

<ExtendTDD> :=          <ExtendFunction> <SubFunction> <LaId> <ReqId> <Error> [<Minor>]
                        [Warning] [<ExtendSubFunctionKeywords>]

<ExtendSubFunctionKeywords> :=
                        -- défini pour chaque service dans l'article approprié de la Partie II

<NationalTDD> :=       <NationalFunction> <SubFunction> <LaId> <ReqId> <Error> [<Minor>]
                        [Warning] [<NationalSubFunctionKeywords>]

<NationalSubFunctionKeywords> :=
                        -- appelle un complément d'étude

<PrivateTDD> :=        <PrivateFunction> <SubFunction> <LaId> <ReqId> <Error> [<Minor>]
                        [Warning] [<PrivateSubFunctionKeywords>]

<PrivateSubFunctionKeywords> :=
                        -- appelle un complément d'étude

```

6.2 Description des éléments TDD

En règle générale, il existe 2 sortes d'éléments de syntaxe dans une description TDD:

- 1) ceux dont les paramètres ne dépendent que de la fonctionnalité de la description TDD;
- 2) ceux dont les paramètres dépendent du service auquel s'applique la description TDD.

Lorsque la description TDD procède à l'envoi ou à la réception de fichiers ou de documents, les paramètres de certains éléments de syntaxe peuvent également être:

- propres au destinataire (à savoir le destinataire principal ou de la copie pour les services de courrier électronique);
- propres au document lui-même (à savoir les formats de transfert et de transmission).

Le Tableau 13 montre les éléments de syntaxe des descriptions TDD *qui ne dépendent pas* du service. Les éléments de syntaxe propres au document, au destinataire ou au service sont décrits dans les articles appropriés de la Partie II de la présente Recommandation.

6.3 Identificateur de code (Code-ID)

Le tout premier octet de chaque description TDD, indépendamment de son codage, indique le système de codage. Cet octet est appelé "Code-ID". Le Tableau 14 donne la liste des identificateurs de type Code-ID qui peuvent être attribués par la présente Recommandation.

6.4 Codage en mode texte

Le présent paragraphe fait référence aux schémas de codage avec l'identificateur Code-ID mis à "A", "B", "I" ou "E".

La syntaxe de type BNF expliquée dans A.1 est utilisée dans les paragraphes suivants pour la description du codage en mode texte.

6.4.1 Syntaxe et formatage

Les éléments de syntaxe répertoriés dans 6.1 sont codés à l'aide de couples mot clé/paramètre. La correspondance des couples mot clé/paramètre et des éléments de syntaxe est montrée ci-dessous dans le Tableau 15. Du fait que certains éléments de syntaxe et la fonctionnalité correspondante dépendent du service de télécommunication qui les prend en charge, seuls seront présentés ceux qui ne dépendent pas du service de télécommunication. Les éléments de syntaxe qui dépendent du service de télécommunication sont décrits dans les articles appropriés de la Partie II de la présente Recommandation.

TABLEAU 13/T.611

Récapitulatif des éléments de syntaxe TDD qui ne dépendent pas du service

Élément de syntaxe	Objectif
<CancelFunction>	Identifie la fonction demandée par la description TDD
<Check>	Donne le format que doit respecter un fichier donné. Ne sert que dans la fonction <CheckTDD>
<CheckFunction>	Identifie la fonction demandée par la description TDD
<ComId>	Représente l'identificateur de communication COMmunication-ID. Voir 5.4.1
<Comment>	Sert à rattacher un commentaire libre à un document, qui sera transmis. Ce commentaire est ensuite enregistré dans l'enregistrement de CA plutôt que transmis avec le document par l'intermédiaire du service de télécommunication
<ConvertFunction>	Identifie la fonction demandée par la description TDD
<CopyFunction>	Identifie la fonction demandée par la description TDD
<Delete>	Annonce à l'application de communication si un document reçu doit être effacé de la mémoire interne de l'application de communication, après récupération réussie par l'application locale appelante
<DeleteFunction>	Identifie la fonction demandée par la description TDD
<Direction>	Spécifie si l'enregistrement de CA appartient à un document entrant ou sortant
<Dispatched>	Spécifie si la fonction <DispatchTDD> a été appliquée à l'enregistrement de CA spécifié
<DispatchFunction>	Identifie la fonction demandée par la description TDD
<Error>	Cet élément de syntaxe renvoie le code d'erreur créé par l'application de communication dans les descriptions TDD de réponse. Il sera vérifié par l'application locale pour le bon fonctionnement
<ExtendFunction>	Identifie la fonction demandée par la description TDD
<ExtendSubFunction-Keywords>	Voir l'article approprié de la Partie II de la présente Recommandation
<FileName>	Contient le chemin d'accès au fichier dans lequel le document est enregistré
<InFormat>	Spécifie le format d'entrée d'un document. Ne sert que pour la fonction <Convert-TDD>
<LaId>	Représente l'identificateur de LA (LA-ID). Voir 5.4.2
<LastTime>	Donne l'heure limite à laquelle l'application de communication doit essayer d'envoyer le document spécifié dans la description TDD correspondante
<Layout>	Définit la structure du fichier cible de la fonction <CopyTDD>
<Minor>	Cet élément de syntaxe renvoie un code d'erreur supplémentaire dans les descriptions TDD de réponse
<NationalFunction>	Identifie la fonction demandée par la description TDD
<NationalSubFunction-Keywords>	Appelle un complément d'étude
<NewLa>	Contient le nouvel identificateur de LA (LA-ID) qui doit servir avec la fonction <DispatchTDD>
<OutFormat>	Spécifie le format de sortie d'un document. Ne sert que dans la fonction <Convert-TDD>
<Previewed>	Spécifie si la fonction <PreviewTDD> a été appliquée à l'enregistrement de CA spécifique

TABLEAU 13/T.611 (*fin*)**Récapitulatif des éléments de syntaxe TDD qui ne dépendent pas du service**

Élément de syntaxe	Objectif
<Printer>	Identificateur de l'imprimante sélectionnée pour l'utilisation avec la fonction <PrintTDD>. Dépend du système d'exploitation. Le fabricant de CA doit décrire dans sa documentation la manière de réaliser l'adressage des imprimantes
<PrintFunction>	Identifie la fonction demandée par la description TDD
<PrivateFunction>	Identifie la fonction demandée par la description TDD
<PrivateSubFunction-Keywords>	Appelle un complément d'étude
<PurgeFunction>	Identifie la fonction demandée par la description TDD
<ReceiveFunction>	Identifie la fonction demandée par la description TDD
<ReqId>	Représente l'identificateur REQ-ID. Voir 5.4.3
<ReceiveTime>	Donne l'heure à laquelle l'application de communication a reçu le document spécifié dans les descriptions TDD correspondantes
<ReqRef>	Représente la référence REQ-REF. Voir 5.4.4
<RescheduleFunction>	Identifie la fonction demandée par la description TDD
<SendackFunction>	Identifie la fonction demandée par la description TDD
<SendFunction>	Identifie la fonction demandée par la description TDD
<SendTime>	Donne l'heure à laquelle l'application de communication doit envoyer le document spécifié dans la description TDD correspondante
<Service>	Spécifie le service de télécommunication utilisé. Voir également la Partie II de la présente Recommandation
<ServiceDependent-KeywordsReceive>	Voir l'article correspondant de la Partie II de la présente Recommandation
<ServiceDependent-KeywordsSend>	Voir l'article correspondant de la Partie II de la présente Recommandation
<ServiceDependent-KeywordsSendack>	Voir l'article correspondant de la Partie II de la présente Recommandation
<State>	Spécifie l'état d'un enregistrement de CA. Voir également le Tableau 4 du 5.6.2
<Status>	Renvoie l'état d'une émission ou d'une réception dans les descriptions TDD de réponse
<SubFunction>	Identifie la fonction demandée par la description TDD
<Target>	Sert dans le <ConvertTDD> et dans le <CopyTDD> pour spécifier le chemin d'accès du fichier de sortie créé par l'application de communication
<TimeRange>	Spécifie un intervalle de temps pour sélectionner des enregistrements de CA donnés
<UserKey>	Contient une clé attachée aux descriptions TDD de demande des fonctions <SendTDD> et <SendAckTDD>
<Warning>	Cet élément de syntaxe renvoie un code d'avertissement supplémentaire dans les descriptions TDD de réponse

TABLEAU 14/T.611

Codage des descriptions TDD – Identificateurs de type Code-ID

Code-ID		Représentation des descriptions TDD
Valeur	Présentation de texte	
41 _{hex}	"A"	Texte lisible, organisé en lignes de couples mot clé/paramètre. Le jeu de caractères utilisé correspond au jeu de caractères ASCII étendu APPLI/COM (voir le Tableau 50)
42 _{hex}	"B"	Texte lisible, organisé en lignes de couples mot clé/paramètre. Le jeu de caractères utilisé correspond à une variante nationale du jeu de caractères défini par la Recommandation UIT-T T.50
43 _{hex}	"C"	Système de codage binaire (présenté en langage C dans la Partie III de la présente Recommandation)
C5 _{hex} ^{a)}	"E"	Texte lisible, organisé en lignes de couples mot clé/paramètre. Le jeu de caractères utilisé correspond au jeu de caractères codés EBCDIC
49 _{hex}	"I"	Texte lisible, organisé en lignes de couples mot clé/paramètre. Le jeu de caractères utilisé correspond au jeu de caractères ASCII normalisés APPLI/COM (voir le Tableau 53)
50 _{hex}	"P"	La présentation et la syntaxe des descriptions TDD sont vues comme étant définies par des règles "privées"
Autres valeurs		Les autres valeurs non répertoriées ci-dessus sont réservées pour la normalisation future
<p>a) Si cet identificateur de type Code-ID doit apparaître dans l'environnement ICE, sa valeur binaire doit être choisie en fonction de la présentation codée choisie pour l'ICE lui-même. Autrement dit, si l'identificateur Code-ID de l'en-tête d'environnement ICE est "I" (ASCII normalisé APPLI/COM) alors la valeur binaire de "E" est codée 45_{hex}.</p>		

TABLEAU 15/T.611

Correspondance des couples mot clé/paramètre pour le codage en mode texte

(Un ↵ indique l'opérateur de formatage de ligne nouvelle)

Élément de syntaxe	Couple mot clé/paramètre
<CancelFunction>	"FUNCTION" ":" "Cancel" ↵
<Check>	"CHECK" ":" <Convert-id-parameter> ↵
<CheckFunction>	"FUNCTION" ":" "Check" ↵
<ComId>	"COMID" ":" <Com-id-parameter> ↵
<Comment>	"COMMENT" ":" <Comment-parameter> ↵
<ConvertFunction>	"FUNCTION" ":" "Convert" ↵
<CopyFunction>	"FUNCTION" ":" "Copy" ↵
<Delete>	"DELETE" ":" <Boolean-parameter> ↵
<DeleteFunction>	"FUNCTION" ":" "Delete" ↵
<Direction>	"DIRECTION" ":" <Direction-parameter> ↵
<Dispatched>	"DISPATCHED" ":" <Boolean-parameter> ↵
<DispatchFunction>	"FUNCTION" ":" "Dispatch" ↵
<Error>	"ERROR" ":" <Error-parameter> ↵

TABLEAU 15/T.611 (fin)

Correspondance des couples mot clé/paramètre pour le codage en mode texte

(Un ¶ indique l'opérateur de formatage de ligne nouvelle)

Elément de syntaxe	Couple mot clé/paramètre
<ExtendFunction>	"FUNCTION" ":" "Extend" ¶
<FileName>	"FILENAME" ":" <Path-parameter> ¶
<InFormat>	"INFORMAT" ":" <Convert-id-parameter> ¶
<LaId>	"LA-ID" ":" <La-id-parameter> ¶
<LastTime>	"LASTTIME" ":" <Date-time-parameter> ¶
<Layout>	"LAYOUT" ":" <Layout-id-parameter> ¶
<Minor>	"MINOR" ":" <Error-parameter> ¶
<NationalFunction>	"FUNCTION" ":" "National" ¶
<NewLa>	"NEWLA" ":" <La-id-parameter> ¶
<OutFormat>	"OUTFORMAT" ":" <Convert-id-parameter> ¶
<Previewed>	"PREVIEWED" ":" <Boolean-parameter> ¶
<Printer>	"PRINTER" ":" <Printer-id-parameter> ¶
<PrintFunction>	"FUNCTION" ":" "Print" ¶
<PrivateFunction>	"FUNCTION" ":" "Private" ¶
<PurgeFunction>	"FUNCTION" ":" "Purge" ¶
<ReceiveFunction>	"FUNCTION" ":" "Receive" ¶
<ReceiveTime>	"RCVTIME" ":" <Date-time-parameter> ¶
<ReqId>	"REQ-ID" ":" <Req-id-parameter> ¶
<ReqRef>	"REQREF" ":" <Req-id-parameter> ¶
<RescheduleFunction>	"FUNCTION" ":" "Reschedule" ¶
<SendAckFunction>	"FUNCTION" ":" "SendAck" ¶
<SendFunction>	"FUNCTION" ":" "Send" ¶
<SendTime>	"SENDTIME" ":" <Send-time-parameter> ¶
<Service>	"SERVICE" ":" <Service-parameter> ¶
<State>	"STATE" ":" <State-parameter> ¶
<Status>	"STATUS" ":" <Status-parameter> ¶
<SubFunction>	"SUBFUNC" ":" <Subfunc-parameter> ¶
<Target>	"TARGET" ":" <Path-parameter> ¶
<TimeRange>	"RANGE" ":" <Date-time-parameter> "," <Date-time-parameter> ¶
<UserKey>	"USERKEY" ":" <Userkey-parameter> ¶
<Warning>	"WARNING" ":" <Error-parameter> ¶

Pour améliorer la lisibilité des descriptions TDD, tous les couples mot clé/paramètre cités doivent être formatés à l'aide d'opérateurs de formatage. Les règles suivantes s'appliquent:

- les éléments de syntaxe sont organisés en lignes de couples mot clé/paramètre;
- le mot clé FUNCTION doit toujours être le premier mot clé; s'il y a un mot clé SUBFUNC pour les descriptions TDD, ce mot clé doit venir en second; il n'est pas défini d'ordre particulier pour les autres mots clés;

- la ligne se termine par un opérateur de formatage de nouvelle ligne (voir le Tableau 16);
- le paramètre doit être séparé du mot clé par deux points (caractère ":");
- chaque ligne peut contenir un commentaire. Celui-ci est introduit par un caractère ";" et va jusqu'en bout de ligne. Le commentaire n'est pas interprété par l'application de communication. Pour inclure un point virgule ";" dans le corps d'une déclaration, il faut le préfixer par une barre oblique inverse (caractère "\");
- une ligne peut contenir des opérateurs de formatage de type espace (voir le Tableau 16);
- les mots clés ne sont pas sensibles à la différence entre majuscules et minuscules (les majuscules et les minuscules sont traitées de la même manière);
- la présentation des paramètres n'est pas non plus sensible à la différence entre majuscules et minuscules, sauf quand le service ou le système d'exploitation qui les prend en charge l'exige.

Une description TDD de demande contient des paramètres d'entrée. La description TDD de réponse peut contenir des paramètres de sortie. L'application de communication doit respecter les règles de construction suivantes pour les descriptions TDD de réponse³⁾:

- tous les paramètres d'entrée de la description TDD de demande doivent être renvoyés dans la description TDD de réponse;
- au moins tous les paramètres de sortie déclarés de base doivent être placés dans la description TDD de réponse par l'application de communication;
- l'ordre des paramètres contenus dans la description TDD de demande peut être changé par l'application de communication à la seule restriction que le mot clé FUNCTION doit être le premier mot clé présenté, suivi du mot clé SUBFUNC s'il y a lieu;
- l'application de communication peut modifier majuscules et minuscules des paramètres contenus dans la description TDD de demande;
- les commentaires contenus dans la description TDD de demande peuvent être supprimés.

TABLEAU 16/T.611

Opérateurs de formatage pour le codage en mode texte

Opérateur de formatage	Codage dépendant de l'identificateur Code-ID			
	"A"	"B"	"E"	"I"
Nouvelle ligne	0D0A _{hex} ou 0A _{hex}	0D0A _{hex} ou 0A _{hex}	Appelle un complément d'étude	0D0A _{hex} ou 0A _{hex}
Espace	20 _{hex} ou 09 _{hex}	20 _{hex} ou 09 _{hex}	Appelle un complément d'étude	20 _{hex} ou 09 _{hex}

Le paragraphe 6.6 explique plus en détail quels paramètres TDD sont utilisés comme paramètres d'entrée et de sortie.

6.4.2 Correspondance des mots clés

6.4.3 Codage de l'en-tête de description TDD

Dans chaque description TDD (et dans l'environnement ICE) l'en-tête <TDD Header> est toujours le premier élément spécifié. Il est structuré de la manière suivante (pour l'explication de la forme de type BNF, voir A.1):

```

<TDD Header> :=      <Code-ID> <Identification> <Version> <Standard> <Reserved>
<Code-ID> :=         "A" | "B" | "E" | "I"
-- pour la présentation de l'identificateur Code-ID, voir également le Tableau 14.

```

³⁾ Ces règles ont considérablement évolué par rapport à la version de 1992 de la présente Recommandation, dont les règles étaient définies de manière beaucoup plus restrictive.

```

<Identification> :=      "*APPLI/COM*"
<Version> :=            "1994"
                        -- l'élément <Version> indique l'année durant laquelle la présente
                        -- Recommandation a été approuvée.
<Standard> :=          "*UIT-T*"
<Reserved> :=          STRING (SIZE(0..16))
                        -- l'élément <Reserved> est réservé à de futures extensions de la présente
                        -- Recommandation.

```

NOTE – L'élément de syntaxe <TDD Header> contient l'identificateur Code-ID comme premier élément.

6.4.4 Codage des paramètres indépendants du service

Le présent paragraphe définit le codage des paramètres indépendants du service, dont il a été question dans le Tableau 15. Les paramètres dépendants du service sont mentionnés dans 6.4.5 et sont spécifiés intégralement dans les paragraphes appropriés de la Partie II de la présente Recommandation.

6.4.4.1 Paramètre booléen

Le paramètre booléen est codé comme une chaîne (STRING) qui peut prendre les valeurs "oui" (YES) ou "non" (NO).

Syntaxe générale:

```

<Boolean-parameter> :=      "YES" | "NO"

```

6.4.4.2 Paramètre Com-id

Le paramètre Com-id est codé comme une chaîne (STRING) contenant l'identificateur de communication Communication-ID (COM-ID) calculé par l'application de communication. Voir également 5.4.1.

Syntaxe générale:

```

<Com-id-parameter> :=      STRING

```

6.4.4.3 Paramètre de commentaire

Le paramètre de commentaire est codé comme une chaîne (STRING) contenant le commentaire spécifié par l'application locale pour un événement de communication donné.

Syntaxe générale:

```

<Comment-parameter> :=      STRING

```

6.4.4.4 Paramètre de date et d'heure

Le paramètre de date et d'heure est codé comme une chaîne (STRING) contenant la date et l'heure avec les conventions "YY-MM-DD-HH:MM" (année-mois-jour:heure-minutes).

Syntaxe générale:

```

<Date-time-parameter> :=      <year> "-" <month> "-" <day> "-" <hours> ":" <minutes>
<year> :=                      <digit> <digit>
<month> :=                     "0" | "1" <digit>
<day> :=                       "0" | ... | "3" <digit>
<hours> :=                     "0" | ... | "2" <digit>
<minutes> :=                   "0" | ... | "5" <digit>
<digit> :=                     "0" | ... | "9"

```

6.4.4.5 Paramètre de direction

Le paramètre de direction est codé comme une chaîne (STRING) qui peut prendre les valeurs "transmission" (XMIT) ou "réception" (RECEIVE).

Syntaxe générale:

<Direction-parameter> := "XMIT" | "RECEIVE"

6.4.4.6 Paramètre d'erreur

Le paramètre d'erreur est codé comme une chaîne (STRING) dont la structure est la suivante:

"nnnn"/"Texte..."

"nnnn" représente un numéro d'erreur à 4 chiffres, justifié à droite, rempli à gauche par des "0". "0000" = succès (aucune erreur). La valeur du numéro d'erreur lui-même dépend du logiciel de télécommunication utilisé. Les codes d'erreur sont décimaux. Pour l'attribution des domaines de code d'erreur, se référer au Tableau 2 du 5.2.3.

"Texte..." représente un texte simple, d'une longueur allant jusqu'à 79 caractères, qui décrit l'erreur. Si le champ de paramètre n'est pas assez long pour accepter du texte simple, ce dernier est abrégé comme nécessaire. Le texte simple pour le numéro d'erreur "0000" est "succès".

Les numéros d'erreur qui peuvent être attribués sont respectivement détaillés dans 5.2.3 et l'Annexe C.

Syntaxe générale:

<Error-parameter> := <digit> <digit> <digit> <digit> / STRING(SIZE(0..79))

<digit> := "0" | ... | "9"

6.4.4.7 Paramètre d'identification de l'application locale

Le paramètre La-id, qui se présente sous la forme d'une chaîne (STRING), contient la référence d'un identificateur de l'application locale (LA-ID). Son intérêt est d'identifier l'application locale "propriétaire" d'une demande.

Syntaxe générale:

<La-id-parameter> := STRING

6.4.4.8 Paramètre d'identification de structure

Le paramètre d'identification de structure est codé comme une chaîne numérique (NUMERIC STRING), qui peut prendre les valeurs "0", "1" ou "2".

Syntaxe générale:

<Layout-id-parameter> := "0" | "1" | "2"

6.4.4.9 Paramètre de chemin

Le paramètre de chemin est codé comme un CHEMIN.

Syntaxe générale:

<Path-parameter> := STRING
-- Limité aux conventions de noms propres à la plate-forme

6.4.4.10 Paramètre d'identification de l'imprimante

Le paramètre d'identification de l'imprimante contient l'identificateur d'une imprimante sélectionnée, codé comme une chaîne (STRING). Le contenu de cette chaîne (STRING) dépend du système d'exploitation qui la prend en charge. Le fabricant de l'application de communication doit préciser dans sa documentation la manière de réaliser l'adressage des imprimantes.

Syntaxe générale:

<Printer-id-parameter> := STRING

6.4.4.11 Paramètre d'identification de demande

Le paramètre d'identification de demande contient la référence d'un identificateur de demande (REQ-ID) (voir 5.4.3). La valeur de paramètre se présente sous forme d'une chaîne (STRING). L'objet de ce paramètre est d'identifier la relation entre une réponse et une demande antérieure. Ainsi, l'identificateur REQ-ID doit être unique dans une même application locale, ce qu'il appartient à l'application locale de vérifier.

Syntaxe générale:

<Req-id-parameter> := STRING

6.4.4.12 Paramètre d'heure d'envoi

Le paramètre d'heure d'envoi est codé comme une chaîne (STRING) contenant la date et l'heure avec les conventions "YY-MM-DD-HH:MM" (année-mois-jour:heure-minutes) ou bien une indication spécifique de traitement immédiat ou urgent.

Syntaxe générale:

<Send-time-parameter> := "IMMEDIATE" | "URGENT" | <Date-time-parameter>
-- Le paramètre <Date-time-parameter> est décrit en 6.4.4.4

6.4.4.13 Paramètre d'état d'enregistrement

Le paramètre d'état spécifie l'état de l'enregistrement de CA. Il est codé comme une chaîne (STRING) et peut prendre les valeurs suivantes:

"retardé" (<i>delayed</i>)	concerne la transmission: l'enregistrement n'a pas encore été traité par l'application de communication.
"en cours d'envoi" (<i>sending</i>)	concerne la transmission: l'enregistrement est traité par l'application de communication pour la transmission.
"envoyé" (<i>sent</i>)	concerne la transmission: l'enregistrement a été transmis avec succès au destinataire.
"échec" (<i>failed</i>)	concerne la transmission et la réception: l'enregistrement n'a pas été transmis entièrement ou des erreurs se sont produites en phase de réception ou une erreur interne à l'application de communication s'est produite.
"réception" (<i>reception</i>)	concerne la réception: l'enregistrement a été reçu mais pas encore récupéré.
"récupéré" (<i>retrieved</i>)	concerne la réception: l'enregistrement a déjà été récupéré.

Syntaxe générale:

<State-parameter> := "delayed" | "sending" | "sent" | "failed" | "reception" | "retrieved"

6.4.4.14 Paramètre d'état d'événement

Le paramètre d'état reflète l'état d'un événement d'émission ou de réception. Il peut prendre les valeurs de chaîne (STRING) suivantes:

- "+" positive;
- "+-" en partie négative;
- "-" négative;
- "?" inconnue.

Syntaxe générale:

<Status-parameter> := "+" | "+-" | "-" | "?"

6.4.4.15 Paramètre de sous-fonction

Le paramètre de sous-fonction est codé comme une chaîne (STRING) qui spécifie la sous-fonction à appeler.

Syntaxe générale:

<Subfunction-parameter> := STRING

6.4.4.16 Paramètre de clé d'utilisateur

Le paramètre de clé d'utilisateur est codé comme une chaîne (STRING) contenant une clé d'utilisateur que l'application locale spécifie pour l'événement de communication donné.

Syntaxe générale:

<Userkey-parameter> := STRING

6.4.5 Codage de paramètres propres au service

Le présent paragraphe décrit certains des paramètres propres au service qui sont utilisés de manière semblable pour divers services. Cependant, une description complète des paramètres mentionnés est donnée dans les articles appropriés de la Partie II de la présente Recommandation.

6.4.5.1 Paramètre d'identification de service

Ce paramètre apparaît dans toutes les descriptions TDD d'émission et de réception. Il spécifie le service à utiliser. Il est codé sous la forme d'une chaîne (STRING) et peut prendre l'une des valeurs suivantes:

"FX3"	pour les services de télécopie du groupe 3
"FX4"	pour les services de télécopie du groupe 4
"TTX"	pour le service télétext
"TX"	pour les services de télex passant par l'intermédiaire du réseau télétext (sans capacité de dialogue)
"TLX"	pour le service télex
"EMAIL"	pour les services de courrier électronique
"FT"	pour les services de transfert de fichier

Syntaxe générale:

<Service-id-parameter> := "FX3" | "FX4" | "TTX" | "TX" | "TLX" | "EMAIL" | "FT"

6.4.5.2 Paramètres File-of-addrspec et paramètre d'adresse

Ces paramètres, qui apparaissent dans toutes les descriptions TDD d'émission et de réception, sont utilisés de manière semblable dans tous les services.

Le paramètre de File-of-addrspec est codé comme un CHEMIN qui pointe sur un fichier contenant les paramètres d'Addrsec, qui en retour contiennent le paramètre d'adresse.

Le paramètre d'adresse est codé sous forme de chaîne (STRING). Pour certains services de télécommunication cette chaîne peut être davantage restreinte.

Syntaxe générale:

<File-of-addrspec> := PATH
-- Chemin vers un fichier contenant une liste de paramètres
-- <Addrsec-parameter>

<Addrsec-parameter> := <Address-parameter> <Other-parameters>
-- Le paramètre <Addrsec-parameter> est entièrement détaillé pour chaque
-- service dans l'article approprié de la Partie II de la présente
-- Recommandation.

<Address-parameter> := -- entièrement décrit dans les articles appropriés de la Partie II de la
-- présente Recommandation.

6.4.5.3 Paramètre Convert-id

Ce paramètre est cité dans diverses descriptions TDD. Certaines d'entre elles, à savoir les descriptions TDD d'émission et de réception, dépendent du service.

Dans tous les cas, le paramètre Convert-id est codé sous forme d'une chaîne (STRING) et contient l'identificateur Convert-ID, qui spécifie le format de transfert. Pour davantage d'informations sur ce paramètre, voir également 5.4.5, l'article 8 et le Tableau H.1, ainsi que les articles appropriés de la Partie II de la présente Recommandation.

Syntaxe générale:

<Convert-id-parameter> := STRING

6.4.5.4 Paramètre Type-id

Ce paramètre, qui est présent dans toutes les descriptions TDD d'émission et de réception, est utilisé de manière similaire pour tous les services.

Le paramètre Type-id est codé comme une chaîne (STRING) et contient l'identificateur de type Type-ID, qui spécifie le format de transmission. Pour davantage d'informations sur le Type-id, voir également 5.4.6 et le Tableau H.2, ainsi que les articles appropriés de la Partie II de la présente Recommandation.

Syntaxe générale:

<Type-id-parameter> := STRING

6.4.5.5 Paramètre File-of-filespec et paramètre de fichier

Ce paramètre, qui apparaît dans toutes les descriptions TDD d'émission et de réception, est utilisé de manière similaire dans tous les services. Le paramètre File-of-filespec est codé comme un CHEMIN qui pointe sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre de fichier.

Le paramètre de fichier est lui-même codé comme un CHEMIN, qui pointe sur le fichier transféré.

Syntaxe générale:

<File-of-filespec> := PATH
-- Chemin vers un fichier contenant une liste de paramètres
-- <Filespec-parameter>

<Filespec-parameter> := <File-parameter> <Other-parameters>
-- Le paramètre <Filespec-parameter> est lui-même entièrement détaillé
-- pour chaque service, dans l'article approprié de la Partie II de la
-- présente Recommandation.

<File-parameter> := PATH
-- Chemin vers le fichier transféré.

6.5 Traitement de documents

6.5.1 Définitions

Certains services de télécommunication permettent de transférer un fichier unique ou un ensemble de fichiers au cours d'un seul événement de communication. Les définitions suivantes sont utiles:

6.5.1.1 document: Ensemble d'informations appartenant à un contexte commun qui sera (ou a été) transmis au cours d'une session par l'intermédiaire d'un service de télécommunication ou de communication sous-jacent. Un document peut comprendre plusieurs fichiers.

6.5.1.2 session: Association entre deux systèmes d'extrémité, leur permettant d'échanger des données sans perte ni interruption visible.

6.5.1.3 file: Compilation de données telle qu'elle est tenue à jour dans le système d'exploitation sous-jacent.

6.5.2 Envoi de documents

Les fonctionnalités suivantes d'envoi de documents sont prises en charge:

- envoyer un document (formant un seul fichier), à un ou à plusieurs destinataires;
- envoyer un document (formant de nombreux fichiers) à un ou à plusieurs destinataires.

Afin d'échanger avec succès un document au moyen d'un service de télécommunication, au moins trois éléments d'information doivent être spécifiés:

- le chemin vers le fichier (ou les chemins vers les fichiers) qui constitue(nt) le document;
- le format de transfert (Convert-ID) dans lequel le document est échangé entre l'application locale et l'application de communication;
- le format de transmission (Type-ID) dans lequel le document est échangé à travers le réseau.

Si le document comprend un fichier, le chemin du fichier peut être donné dans l'élément de syntaxe <Document>, le format de transmission dans l'élément de syntaxe <Type> et le format de transfert dans l'élément de syntaxe <Convert>. Du fait que les valeurs de ces éléments de syntaxe dépendent du service, ils sont décrits pour chaque service dans la Partie II de la présente Recommandation.

Si le document comprend plusieurs fichiers, les 3 composantes doivent être rassemblées conformément à une syntaxe propre au service. Cette syntaxe étant propre au service, elle est également décrite pour chaque service dans la Partie II de la présente Recommandation. D'après cette syntaxe, les composantes rassemblées doivent être mises dans un fichier à adressage indirect par l'application locale. Le chemin de ce fichier à adressage indirect doit ensuite être spécifié dans l'élément de syntaxe approprié de la description TDD.

NOTE – Le chemin d'un fichier à adressage indirect est toujours spécifié avec un caractère "@" en tête!

6.5.3 Réception de documents

Les fonctionnalités suivantes sont assurées, en ce qui concerne la réception de documents:

- recevoir un document formant un seul fichier;
- recevoir un document formant de nombreux fichiers.

Si un document reçu forme de nombreux fichiers, l'application de communication construit un fichier à adressage indirect tel que décrit dans 6.5.2 et met un caractère @ avant le chemin de ce fichier.

Si le document reçu forme un seul fichier, le chemin (facultativement) donné par l'application locale doit être utilisé pour mémoriser le document lui-même. Le format de transmission est ensuite renvoyé dans l'élément de syntaxe <Type> et le format de transfert dans l'élément de syntaxe <Convert>.

6.5.4 Format du fichier à adressage indirect

Le fichier à adressage indirect comprend des éléments de syntaxe propres au service à raison d'une ligne par fichier. Le code de présentation doit se conformer à l'identificateur Code-ID utilisé pour les descriptions TDD. Se référer au 6.4.5.5 pour une description générique des éléments de syntaxe contenus dans le fichier à adressage indirect.

NOTE – Le fichier à adressage indirect doit être traité par ordre d'apparition.

6.6 Fonctionnalité des descriptions TDD

Le présent paragraphe décrit la fonctionnalité qui est déclenchée par les descriptions TDD de demande. Il donne des informations sur la manière d'utiliser les éléments de syntaxe dans la demande et dans la réponse. Le Tableau 17 explique les titres des colonnes des tableaux de descriptions TDD utilisés dans les paragraphes qui suivent.

TABLEAU 17/T.611

Explication des titres de colonne pour les tableaux de description TDD

Colonne	Contenu
Élément de syntaxe	Donne la liste des éléments de syntaxe utilisés pour la description TDD
C (classe)	Indique la classe du mot clé. "B" est mis pour "de base" et signifie que le mot clé doit être accepté par toutes les applications de communication. "+" signifie que le mot clé est accepté par l'application de communication si et seulement si le mot clé est déclaré dans l'ICE
T (type)	Indique si le mot clé est obligatoire ("m") ou à option ("o"). Lorsque le mot clé est à option, la cellule "valeur par défaut" indique une valeur par défaut pour le paramètre
I/O (entrée/sortie)	<p>La colonne d'entrée/sortie indique les conditions requises concernant le paramètre qui suit le mot clé dans la demande et la réponse</p> <p>Lorsque la colonne indique "I", cela signifie que le paramètre contient une valeur d'entrée, c'est-à-dire que l'application de communication doit interpréter la valeur comme étant celle qui est demandée par l'application locale. L'application de communication ne doit pas modifier la valeur et doit la renvoyer dans la réponse</p> <p>Lorsque la colonne indique "I/O", cela signifie que le paramètre contient une valeur d'entrée (comme lorsque "I" est spécifié) et qu'une valeur de "sortie" (c'est-à-dire fixée par l'application de communication dans la description TDD de réponse) est attendue</p> <p>Lorsque la colonne indique "O", cela signifie que le paramètre est une valeur de sortie (c'est-à-dire fixée par l'application de communication dans la description TDD de réponse) et qu'aucune valeur ne doit être fixée dans la demande (c'est-à-dire qu'aucune valeur de paramètre ne doit être spécifiée dans la demande). La réponse doit contenir la valeur utilisée pour ce paramètre par l'application de communication</p>
Mot clé	Indique le nom du mot clé tel qu'il figure dans la description TDD lorsqu'elle est codée en mode texte. Les majuscules peuvent être utilisées aussi bien que les minuscules; par exemple les mots "Fonction" et "funcTion" doivent être interprétés de la même façon. Voir également 6.4.1
Paramètre	Indique les valeurs possibles pour le paramètre lorsqu'il est codé en mode texte. La correspondance avec les types de paramètres est donnée au 6.4.4
Valeur par défaut	Indique la valeur par défaut pour le paramètre (s'il y a lieu, c'est-à-dire lorsque la cellule "Type" indique "O")
Commentaire	Donne une brève explication pour l'utilisation des mots clés et des paramètres

Les règles générales à utiliser sont énoncées ci-après:

- une description TDD donnée (demande et réponse) appartient à une classe fonctionnelle. En conséquence, les mots clés associés ne dépendent pas de la classe fonctionnelle;
- les mots clés sont divisés en deux catégories: "de base" et additif ("+"). Les mots clés additifs ne peuvent être utilisés par les applications locales que s'ils sont déclarés dans l'ICE (descripteur de CA). Les mots clés de base sont acceptés par toutes les applications de communication;
- l'utilisation ou la non-utilisation d'un mot clé a la même signification pour les mots clés de base et pour les mots clés additifs;
- si un mot clé est absent dans une description TDD, la valeur par défaut du paramètre s'applique. Une valeur par défaut est toujours un paramètre d'entrée, c'est-à-dire que la classification d'"entrée/sortie" est toujours "I";
- lorsqu'un paramètre est classé "I" (entrée), l'application de communication ne doit pas modifier la valeur dans la description TDD de réponse.

Le Tableau 18 donne des explications sur la manière dont les classifications de type et d'entrée/sortie sont utilisées:

TABLEAU 18/T.611

Utilisation des classifications de type et d'entrée/sortie

Type	Entrée/sortie	Le mot clé
"m"	"I"	Doit être spécifié dans la demande; une valeur de paramètre doit aussi être spécifiée. Dans la réponse, le mot clé est également spécifié, la valeur de paramètre restant inchangée
	"O"	Doit être spécifié dans la demande avec une valeur de paramètre vide. Dans la réponse, le mot clé est également présent avec une valeur de paramètre significative
	"I/O"	Doit être spécifié dans la demande; la valeur de paramètre doit aussi être spécifiée. Dans la réponse, le mot clé est aussi spécifié, avec une valeur de paramètre qui peut être différente. La valeur de paramètre contenue dans la réponse est significative
"o"	"I"	Dans la demande, le mot clé peut être présent ou non; s'il est absent, la valeur par défaut s'applique. La réponse ne peut inclure le mot clé que s'il est spécifié dans la demande; en ce cas, la valeur de paramètre reste inchangée
	"O"	Peut être spécifié dans la demande (sans aucune valeur de paramètre) si le but recherché est d'obtenir une valeur de paramètre réponse pour le mot clé; en ce cas, une valeur de paramètre doit être spécifiée dans la réponse. Dans le cas contraire, aucune valeur de paramètre n'est renvoyée. Le mot clé n'a pas de valeur par défaut
	"I/O"	Peut être spécifié dans la demande avec une valeur de paramètre. Si le mot clé n'est pas spécifié dans la demande, il prend sa valeur de paramètre par défaut. Dans la réponse, le mot clé n'est présent que s'il était présent dans la demande; en ce cas, la valeur de paramètre réponse est importante. Si la demande ne spécifie pas le mot clé, la réponse ne peut pas non plus le spécifier

6.6.1 Send (envoi)

Les fonctions <SendTDD> et <SendAckTDD> sont utilisées pour envoyer un ou plusieurs documents à un ou à plusieurs destinataires. La réponse <SendAckTDD> sert d'accusé de réception d'une demande antérieure. Pour la demande <SendTDD>, aucune réponse n'est créée par l'application de communication.

La fonction <SendTDD> peut être utilisée pour envoyer des documents à un nombre quelconque de destinataires. La fonction <SendAckTDD> n'autorise qu'un seul destinataire par demande. La fonction <SendAckTDD> est conçue pour les applications locales ne mettant en œuvre que la classe fonctionnelle A; pour ces applications, l'application locale doit donc produire un rapport d'achèvement. La fonction <SendTDD> est conçue pour les applications locales mettant en œuvre la classe fonctionnelle B; ces applications sont donc à même de suivre les descriptions TDD de demande au moyen de la fonction Trace. Les deux types d'applications locales peuvent toutefois utiliser les fonctions <SendTDD> et <SendAckTDD>.

Si une application locale veut envoyer des documents à une liste de destinataires, elle doit utiliser la fonction <SendTDD> avec l'élément de syntaxe <Addrlist>. L'état d'émission (élément de syntaxe <Status>) n'est pas disponible par l'intermédiaire de cette fonction, du fait qu'aucune description TDD de réponse n'est créée. Cependant, l'état d'émission peut être obtenu en utilisant la fonctionnalité Trace:COPY, telle que décrite dans 6.6.3.

NOTE – La fonctionnalité Trace n'est disponible qu'à travers les applications de communication qui sont conformes à la classe fonctionnelle B (FCB).

Deux tableaux sont fournis: l'un pour la situation <SendTDD>, l'autre pour la situation <SendAckTDD>.

La réponse <SendAckTDD> est toujours interprétée comme un état d'achèvement, c'est-à-dire que si la demande aboutit, la réponse doit être créée lorsque le traitement de la demande <SendAckTDD> est achevé. Aussi, dès qu'une description TDD de réponse est disponible, l'application locale peut se fonder explicitement sur le contenu du paramètre d'état montrant le résultat d'émission. Le paramètre d'état indique l'un des 4 états suivants:

- 1) document entièrement émis avec succès;
- 2) échec de l'émission, tous les essais d'émission sont abandonnés par l'application de communication;

- 3) échec partiel de l'émission: certaines parties sont déjà émises, mais aucune émission complète n'a pu être effectuée;
- 4) état inconnu.

Les fonctions <SendTDD> et <SendAckTDD> appartiennent aux classes fonctionnelles A et B.

Si une demande <SendAckTDD> mentionne expressément plusieurs destinataires, les éléments de syntaxe <Status> et <Error> ne peuvent pas être spécifiés (voir les Tableaux 19 et 20).

TABLEAU 19/T.611

Éléments de syntaxe de la fonction <SendAckTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<SendAckFunction>	B	m	I	FUNCTION	"SendAck"	–	L'application de communication doit créer une réponse
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqID>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Service>	B	m	I	SERVICE	<Service-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Status>	B	m	O	STATUS	<Status-parameter>	–	Renvoi de l'état par l'application de communication
<ComId>	B	o	O	COMID	<Com-id-parameter>	–	Identification de la communication (calculée par l'application de communication)
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	L'application de communication doit effectivement traiter la demande à l'heure spécifiée
<Comment>	+	o	I	COMMENT	<Comment-parameter>	–	
<LastTime>	+	o	I	LASTTIME	<Date-time-parameter>	Dépend de la CA	Heure limite pour traiter la demande, exprimée en temps absolu
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<UserKey>	+	o	I	USERKEY	<Userkey-parameter>	–	La demande et la réponse doivent contenir la même valeur de paramètre. L'application de communication ne doit absolument pas interpréter ce paramètre
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

TABLEAU 20/T.611

Éléments de syntaxe de la fonction <SendTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<SendFunction>	B	m	I	FUNCTION	"Send"	–	L'application de communication ne doit PAS créer de réponse
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Service>	B	m	I	SERVICE	<Service-id-parameter>	–	
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	L'application de communication doit effectivement traiter la demande à l'heure spécifiée
<Comment>	+	o	I	COMMENT	STRING	–	
<LastTime>	+	o	I	LASTTIME	<Date-Time-parameter>	Dépend de la CA	Heure limite pour traiter la demande, exprimée en temps absolu
<UserKey>	+	o	I	USERKEY	STRING	–	L'application de communication ne doit absolument pas interpréter ce paramètre

6.6.2 Receive (réception)

Les demandes <ReceiveTDD> demandent la récupération d'un document entrant déjà reçu par l'application de communication. La fonction <ReceiveTDD> appartient aux classes fonctionnelles A et B (voir le Tableau 21).

La demande spécifie le service de télécommunication, la zone mémoire pour le fichier entrant et le format de transfert désiré. Pour certains services, la sous-adresse du destinataire peut également servir de sélecteur (voir l'article approprié de la Partie II de la présente Recommandation).

Afin de récupérer (recevoir) un document, l'application locale peut utiliser deux méthodes différentes:

- soit l'application locale récupère tout ce qui se présente par la suite sans aucune présélection; dans ce cas, elle doit être capable de traiter convenablement le document reçu, indépendamment de son format de transfert;
- soit l'application locale commence par inspecter une copie d'une liste de documents reçus en utilisant la demande Trace <CopyTDD>. Avec cette liste, elle peut obtenir l'identificateur COM-ID du document qu'elle souhaite récupérer. Cela n'est possible que si l'application de communication est de type classe fonctionnelle B et que l'application locale assure la fonction <CopyTDD>.

L'élément de syntaxe <Convert> sert à enregistrer le format de transfert réel du document à l'intérieur de la description TDD de réponse; les éléments de syntaxe <Status> et <Error> sont nécessaires pour acheminer l'état et le code d'erreur de l'événement de réception. Si l'application locale connaît l'identificateur COM-ID du document qu'elle recherche, elle peut aussi spécifier l'élément de syntaxe <ComId> dans la demande afin d'extraire ce document.

L'élément de syntaxe *additif* <Delete> commande la suppression du document à l'intérieur de l'application de communication. Pour un fonctionnement normal, l'élément <Delete> doit être vrai et l'application de communication doit effacer le document de sa mémoire après transfert. Aussi un document ne peut-il être reçu qu'une fois, forçant à l'enregistrement de CA à l'état "récupéré", comme le décrit 5.6.7.1.

Si l'élément <Delete> est mis à faux, le document sera conservé par l'application de communication et l'enregistrement de CA restera en état de réception. Aussi, à la prochaine tentative, le même document pourra-t-il être récupéré de nouveau.

NOTE – L'instance d'application de communication peut décider de ne procéder à l'effacement que virtuellement et peut conserver les documents "effacés" de manière interne, par exemple pour les rendre disponibles aux autres applications locales ou pour les conserver à des fins d'archivage.

TABLEAU 21/T.611

Éléments de syntaxe de la fonction <ReceiveTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<ReceiveFunction>	B	m	I	FUNCTION	"Receive"	–	
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Status>	B	m	O	STATUS	<Status-parameter>	–	Renvoie l'état de l'application de communication. Quand aucun document n'est disponible, la description TDD de réponse doit spécifier la valeur "–"
<ComId>	B	o	I/O	COMID	<Com-id-parameter>	–	Identification de la communication (calculée par l'application de communication)
<Service>	B	o	I/O	SERVICE	<Service-id-parameter>	–	S'il figure dans la demande, cet élément sert de critère de sélection
<Delete>	+	o	I	DELETE	<Boolean-parameter>	"Oui"	Si la valeur est "vrai", l'application de communication doit effacer le document récupéré de sa mémoire tampon interne
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<ReceiveTime>	+	o	O	RCVTIME	<Date-time-parameter>	–	Heure de la réception du document par l'application de communication
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

6.6.3 Trace (suivi)

Les descriptions TDD du groupe Trace servent à gérer les enregistrements de CA. La fonctionnalité Trace est propre à la classe fonctionnelle B.

Les diverses descriptions TDD de ce groupe ont pour objet de gérer le fonctionnement d'une application de communication. Les descriptions TDD définies sont les suivantes:

- <PurgeTDD> est utilisée pour SUPPRIMER tout enregistrement de CA qui se trouve dans l'état "échec", "récupéré" ou "envoyé"; cette fonction est utile pour libérer les enregistrements de CA ayant atteint un état d'inactivité;
- <CopyTDD> est utilisée pour COPIER tout enregistrement de CA, quel que soit l'état dans lequel il se trouve, dans un fichier; cette fonction est utile pour construire ce que l'on désigne communément sous le nom de "journaux";
- <CancelTDD> est utilisée pour ANNULER tout enregistrement de CA à l'état "en cours d'envoi"; cette fonction permet à l'application locale d'interrompre la transmission que l'application de communication est en train de réaliser⁴⁾ et d'y mettre fin;
- <DeleteTDD> est utilisée pour EFFACER tout enregistrement de CA à l'état "retardé" afin de l'annuler;
- <RescheduleTDD> est utilisée pour REPROGRAMMER tout enregistrement de CA à l'état "échec" pour permettre à l'application de communication de le retransmettre; cette fonction facilite la gestion des transmissions inefficaces pour cause d'occupation de la ligne du destinataire ou de mauvaise formulation de la demande, par exemple;
- <DispatchTDD> est utilisée pour DISTRIBUER tout enregistrement de CA à l'état "réception" pour l'affecter à l'application locale destinataire effective;
- <PreviewTDD> est utilisée pour PRÉVISUALISER un document associé à un enregistrement de CA dans l'état de "réception".

Il est à noter que toutes les demandes sont exécutées au nom d'une application locale donnée (par l'intermédiaire de l'identificateur LA-ID). Cela limite le champ des fonctions ci-dessus aux enregistrements de CA affectés à cet identificateur LA-ID (sauf pour les situations évidentes où l'identificateur LA-ID n'est pas spécifié).

Pour effacer une demande antérieure dans l'état "en cours d'envoi", la fonction <DeleteTDD> doit être utilisée.

L'application locale peut spécifier l'élément de syntaxe <ComId>, s'il est connu, au lieu de l'élément de syntaxe <ReqRef>.

6.6.3.1 Trace:DELETE

Voir Tableau 22.

6.6.3.2 Trace:COPY

La fonction <CopyTDD> est utilisée pour obtenir une copie des enregistrements de CA qui font référence à un état donné.

L'élément de syntaxe <State> spécifie l'état d'un enregistrement de CA que l'on a l'intention de copier. L'élément de syntaxe <Target> spécifie le chemin d'accès au fichier dans lequel l'application de communication émettra son paramètre de sortie. L'élément de syntaxe <Error> sert à enregistrer le code d'erreur de l'opération dans la description TDD de réponse. La liste copiée est toujours dans le format imposé par l'identificateur Code-ID de la description TDD qui est à l'origine de la demande. La structure de la liste est imposée par la valeur du champ RECORD (enregistrement) de l'ICE. L'ordre des champs dans la liste cible doit être le même que l'ordre dans lequel les mots clés sont déclarés dans le champ RECORD.

L'application locale peut spécifier l'élément de syntaxe <ComId>, s'il est connu, au lieu de l'élément de syntaxe <ReqRef>.

L'élément de syntaxe facultatif <Layout> peut être utilisé afin de spécifier la structure de la copie résultante. Les valeurs que ce paramètre peut prendre sont explicitées dans le Tableau 23, les Figures 7 et 8 en donnant des exemples. La représentation en mode caractère de la copie est imposée par l'en-tête <TDDHeader> de la fonction <CopyTDD> correspondante (voir le Tableau 24).

⁴⁾ Rien ne garantit que l'opération annulation réussira, en raison de la nature de l'interface.

TABLEAU 22/T.611

Éléments de syntaxe de la fonction <DeleteTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<DeleteFunction>	B	m	I	FUNCTION	"Delete"	–	Efface l'enregistrement de CA dans l'état "retardé"
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilisation de l'élément de syntaxe <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identification de la communication calculée par l'application de communication
Utilisation de l'élément de syntaxe <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Référence à un identificateur REQ-ID antérieur

TABLEAU 23/T.611

Impact de l'élément de syntaxe <Layout>

Layout-id (identification de la structure)	Explication
0	Valeur par défaut. Le nombre, la structure et l'ordre des champs présentés dans la cible copie résultante sont définis par l'ICE dans le champ RECORD (enregistrement). Le champ RECORD donne la longueur des champs et impose leur ordre. Les champs d'un enregistrement de CA se présentent sur une ligne, remplie avec des SPACE (espaces) à la longueur spécifiée dans le champ RECORD. Il n'est pas fourni d'en-tête
1	Les champs se présentent ligne par ligne comme des valeurs délimitées par des virgules (csv), incluses dans des apostrophes. La première ligne d'un champ contient les mots clés auxquels la colonne de champ se réfère. Voir également la Figure 7
2	Les champs se présentent ligne par ligne, délimités par des caractères de tabulation. La première ligne d'un champ contient les mots clés auxquels la colonne de champ se réfère. Voir également la Figure 8

```
"COMID","DIRECTION","LA-ID","REQ-ID","STATE"  
"0001","émission","Jean","REQ-1212","retardé"  
"0002","émission","Jean","REQ-1213","envoyé"  
"0003","réception","Jean","REQ-1214","récupéré"
```

FIGURE 7/T.611

Exemple de structure avec des valeurs délimitées par des virgules (Layout-id = 1)

COMID	DIRECTION	LA-ID	REQ-ID	STATE
0001	émission	Jean	REQ-1212	retardé
0002	émission	Jean	REQ-1213	envoyé
0003	réception	Jean	REQ-1214	récupéré

FIGURE 8/T.611

Exemple de structure avec des valeurs délimitées par des tabulations (Layout-id = 2)

TABLEAU 24/T.611

Éléments de syntaxe de la fonction <CopyTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<CopyFunction>	B	m	I	FUNCTION	"Copy"	–	Copie tous les enregistrements de CA d'un état donné dans le fichier sur lequel pointe le mot clé TARGET
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<State>	B	m	I	STATE	<State-parameter>	Tous états	Etat de l'enregistrement de CA
<Target>	B	m	I	TARGET	<Path-parameter>	–	Nom de fichier de destination
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Layout>	B	o	I	LAYOUT	<Layout-id-parameter>	"0"	Spécifie la structure de la copie résultante
<Direction>	B	o	I	DIRECTION	<Direction-parameter>	Dans les deux sens	
<Dispatched>	B	o	I	DISPATCHED	<Boolean-parameter>	Tous enreg. de CA	
<Previewed>	B	o	I	PREVIEWED	<Boolean-parameter>	Tous enreg. de CA	
<Service>	B	o	I	SERVICE	<Service-id-parameter>	Tous services	
<TimeRange>	B	o	I	RANGE	<Date-time-parameters>	Tous enreg. de CA	
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilisation de l'élément de syntaxe <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identification de la communication calculée par l'application de communication
Utilisation de l'élément de syntaxe <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Référence à un identificateur REQ-ID antérieur

6.6.3.3 Trace: CANCEL

Pour annuler une demande "d'envoi" antérieure, l'application locale doit utiliser la fonction <CancelTDD> (voir le Tableau 25).

L'application locale peut également spécifier l'élément de syntaxe <ComId>, s'il est connu, au lieu de l'élément de syntaxe <ReqRef>.

NOTE – L'annulation d'une demande d'envoi peut échouer en raison de la nature de l'interface.

TABLEAU 25/T.611

Éléments de syntaxe de la fonction <CancelTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<CancelFunction>	B	m	I	FUNCTION	"Cancel"	–	Annule un enregistrement de CA créé par une demande antérieure
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilisation de l'élément de syntaxe <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identification de la communication calculée par l'application de communication
Utilisation de l'élément de syntaxe <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Référence à un identificateur REQ-ID antérieur

6.6.3.4 Trace: PURGE

La fonction <PurgeTDD> est utilisée afin de supprimer les enregistrements de l'application de communication (voir le Tableau 26).

L'élément de syntaxe <State> spécifie l'état des enregistrements de CA à supprimer (voir 5.6). L'élément de syntaxe <Error> enregistre le code d'erreur de l'opération dans la description TDD de réponse.

L'application locale peut également spécifier l'élément de syntaxe <ComId>, s'il est connu, au lieu de l'élément de syntaxe <ReqRef>.

TABLEAU 26/T.611

Éléments de syntaxe de la fonction <PurgeTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<PurgeFunction>	B	m	I	FUNCTION	"Purge"	–	Efface un enregistrement de CA créé par une demande antérieure
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<State>	B	m	I	STATE	<State-parameter>	Tous états ^{a)}	Etat de l'enregistrement de CA
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilisation de l'élément de syntaxe <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identification de la communication calculée par l'application de communication
Utilisation de l'élément de syntaxe <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Référence à un identificateur REQ-ID antérieur
^{a)} "Tous états", dans le cas présent, représente les états "envoyé", "échec" et "récupéré". L'application de communication doit ignorer l'application de la fonction de "purge" à d'autres états.							

6.6.3.5 Trace:RESCHEDULE

La fonction <RescheduleTDD> est utilisée pour reprogrammer une demande d'envoi qui a échoué. L'application locale peut également spécifier l'élément de syntaxe <ComId>, s'il est connu, au lieu de l'élément de syntaxe <ReqRef> (voir le Tableau 27).

TABLEAU 27/T.611

Éléments de syntaxe de la fonction <RescheduleTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<Reschedule-Function>	B	m	I	FUNCTION	"Reschedule"	–	Reprogramme un enregistrement de CA créé par une demande d'envoi antérieure
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Address>	B	o	I	ADDRESS	<Address-parameter>	–	Sert uniquement pour la fonction "reschedule" (reprogrammation); spécifie la nouvelle adresse du destinataire
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	Traite la demande à l'heure spécifiée
<LastTime>	+	o	I	LASTTIME	<Date-time-parameter>	Dépend de la CA	Heure limite pour le traitement de la demande
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilisation de l'élément de syntaxe <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identification de la communication calculée par l'application de communication
Utilisation de l'élément de syntaxe <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Référence à un identificateur REQ-ID antérieur

6.6.3.6 Trace:DISPATCH

La fonction <DispatchTDD> est utilisée pour distribuer un fichier reçu à une application locale (voir le Tableau 28).

L'élément de syntaxe <NewLa>, qui peut être spécifié à plusieurs reprises, contient l'identificateur LA-ID de la nouvelle application locale. Après un renvoi efficace, le document reçu appartient à la nouvelle application locale et celle-ci peut le récupérer.

NOTE – L'utilisation de la fonction de distribution peut être limitée.

TABLEAU 28/T.611

Eléments de syntaxe de la fonction <DispatchTDD>

Elément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<DispatchFunction>	B	m	I	FUNCTION	"Dispatch"	–	Distribue les documents associés à l'enregistrement de CA désigné par l'identificateur COM-ID
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<NewLa>+	B	m	I	NEWLA	<La-id-parameter>	–	Spécifie le nom du nouveau "propriétaire" du document. Peut être spécifié à plusieurs reprises
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identification de la communication calculée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

6.6.3.7 Trace:PREVIEW

La fonction <PreviewTDD> permet à un administrateur d'application de communication de récupérer et de distribuer les documents reçus, tout en les gardant encore à disposition (pour des distributions ultérieures). La fonction <PREVIEW> fonctionne comme la fonction Receive (réception), à la différence près que l'enregistrement de CA reste dans l'état de réception au lieu de passer dans l'état "reçu" (voir le Tableau 29).

L'enregistrement de CA doit également enregistrer si le document reçu a été vu par l'administrateur et si le document a déjà été distribué. Cela permet à une fonction <Trace:COPY> de rechercher les documents reçus qui n'ont pas déjà été vus ou distribués à leurs destinataires ultimes.

Les principes suivants s'appliquent également:

Lorsque après réception d'un document entrant, l'application de communication crée l'enregistrement de CA (appelé, aux fins de la présente explication, enregistrement "primaire" de CA), ce dernier est affecté à l'utilisateur "administrateur". Les indicateurs "vu" et "distribué" de l'enregistrement primaire de CA sont mis à "non".

Pour distribuer le document entrant, l'administrateur "prévisualise" le document grâce à la fonction <Trace:PREVIEW>. L'indicateur "vu" de l'enregistrement primaire de CA est alors mis à "oui". Ensuite, l'administrateur distribue le document à un ou plusieurs destinataires et l'indicateur "distribué" de l'enregistrement primaire de CA est alors mis à "oui".

L'application de communication crée de manière interne une nouvelle copie de l'enregistrement primaire de CA pour chaque destinataire avec les attributs suivants:

- état = Réception
- LA-ID = LoginName (nom d'ouverture de session) du destinataire auquel elle a été distribuée
- vu = "non"
- distribué = "non"

Cet enregistrement de CA est appelé enregistrement de CA "secondaire" pour des raisons de clarté. Chaque enregistrement de CA "secondaire" appartient désormais au destinataire auquel il a été distribué, qui peut le récupérer grâce à la fonction <Trace:COPY> habituelle et qui peut le recevoir grâce à la fonction Receive. Si le système l'autorise, l'utilisateur peut également distribuer ou prévisualiser le document reçu grâce aux mêmes appels de fonction.

Ce mécanisme permet à l'administrateur de l'application de communication d'établir une liste de tous les destinataires d'un document donné, d'observer qui les a récupérés, de redistribuer un document sur demande, etc.

TABLEAU 29/T.611

Éléments de syntaxe de la fonction <PreviewTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<PreviewFunction>	B	m	I	FUNCTION	"Preview"	-	Récupère des documents associés à l'enregistrement de CA désigné par l'identificateur COM-ID
<LaId>	B	m	I	LA-ID	<La-id-parameter>	-	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	-	
<ComId>	B	m	I	COMID	<Com-id-parameter>	-	Identification de la communication calculée par l'application de communication
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	-	Format de transfert du fichier cible
<Target>	B	m	I	TARGET	<Path-parameter>	-	Nom du fichier cible
<Error>	B	m	O	ERROR	<Error-parameter>	-	Erreur renvoyée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	-	
<Warning>	+	o	O	WARNING	<Error-parameter>	-	

6.6.4 Submit (soumission)

Le groupe de fonctions de type "soumission" a été étudié pour permettre à l'application de communication d'assurer diverses fonctions utilitaires.

Son but est de demander à l'application de communication d'assurer des fonctions supplémentaires relevant de ses capacités, comme les conversions de format de fichier ou l'impression des documents entrants. Normalement, ces tâches ne sont pas accomplies par l'application de communication, mais certains fabricants peuvent souhaiter les assurer. A titre d'exemple de situation où la caractéristique d'"impression" peut être utile, on peut citer la situation d'un "serveur d'application de communication" sur un réseau de zone locale (RZL). Les descriptions TDD suivantes forment le groupe "soumission":

- <PrintTDD> pour IMPRIMER un document, connaissant son chemin d'accès et son format;
- <ConvertTDD> pour CONVERTIR un document, connaissant son chemin d'accès, son format d'entrée, son format de sortie et le nom de son fichier de sortie;
- <CheckTDD> pour VÉRIFIER le format de transfert d'un document, connaissant son chemin d'accès et le format de transfert par rapport auquel il doit être vérifié.

6.6.4.1 Submit: PRINT

Voir le Tableau 30.

TABLEAU 30/T.611

Eléments de syntaxe de la fonction <PrintTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<PrintFunction>	B	m	I	FUNCTION	"Print"	-	Soumet la tâche d'impression à l'application de communication
<LaId>	B	m	I	LA-ID	<La-id-parameter>	-	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	-	
<FileName>	B	m	I	FILENAME	<Path-parameter>	-	Fichier original
<InFormat>	B	m	I	INFORMAT	<Convert-id-parameter>	-	Format original
<Error>	B	m	O	ERROR	<Error-parameter>	-	Erreur renvoyée par l'application de communication
<Printer>	+	o	I	PRINTER	<Printer-id-parameter>	"STD"	La valeur "STD" désigne l'imprimante type configurée dans l'application de communication. D'autres valeurs du paramètre Printer-id peuvent être déclarées pour l'application de communication dans la composante "PRINT" de l'ICE (voir 9.5)
<Minor>	+	o	O	MINOR	<Error-parameter>	-	
<Warning>	+	o	O	WARNING	<Error-parameter>	-	

6.6.4.2 Submit: CONVERT

Voir le Tableau 31.

TABLEAU 31/T.611

Éléments de syntaxe de la fonction <ConvertTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<ConvertFunction>	B	m	I	FUNCTION	"Convert"	–	Soumet la tâche de conversion à l'application de communication
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<FileName>	B	m	I	FILENAME	<Path-parameter>	–	Fichier original
<Target>	B	m	I	TARGET	<Path-parameter>	–	Fichier cible
<InFormat>	B	m	I	INFORMAT	<Convert-id-parameter>	–	Format original
<OutFormat>	B	m	I	OUTFORMAT	<Convert-id-parameter>	–	Format cible
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

6.6.4.3 Submit: CHECK

Voir le Tableau 32.

TABLEAU 32/T.611

Éléments de syntaxe de la fonction <CheckTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<DeleteFunction>	B	m	I	FUNCTION	"Check"	–	Soumet la tâche de vérification à l'application de communication
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	

TABLEAU 32/T.611 (*fin*)

Éléments de syntaxe de la fonction <CheckTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<FileName>	B	m	I	FILENAME	<Path-parameter>	–	Fichier original
<Check>	B	m	I	CHECK	<Convert-id-parameter>	–	Format pour la vérification
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

6.6.5 Extend

La fonction "extend" (extension) se répartit en trois classes, comme suit:

- <ExtendTDD> pour ÉTENDRE la fonction sur une base générale;
- <NationalTDD> pour étendre la fonction sur une base NATIONALE;
- <PrivateTDD> pour étendre la fonction sur une base PRIVÉE.

6.6.5.1 Extend: EXTEND

La fonction <ExtendTDD> est conçue pour offrir des caractéristiques "étendues", qui ne sont pas fondamentales pour le fonctionnement de l'interface, mais dont l'utilisation se généralise ou s'impose. La fonction <ExtendTDD> est un service complémentaire (voir le Tableau 33).

TABLEAU 33/T.611

Éléments de syntaxe minimaux de la fonction <ExtendTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<ExtendFunction>	B	m	I	FUNCTION	"Extend"	–	
<SubFunction>	B	m	I	SUBFUNC	<Subfunction-parameter>	–	Pour l'affectation de valeurs, voir l'article approprié de la Partie II de la présente Recommandation
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication

6.6.5.2 Extend: NATIONAL

La fonction <NationalTDD> a pour but d'offrir des caractéristiques "nationales" propres à chaque pays. La fonction <NationalTDD> est un service complémentaire (voir le Tableau 34).

TABLEAU 34/T.611

Éléments de syntaxe minimaux de la fonction <NationalTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<NationalFunction>	B	m	I	FUNCTION	"National"	–	
<SubFunction>	B	m	I	SUBFUNC	<Subfunction-parameter>	–	Les valeurs de ce paramètre feront l'objet d'un complément d'étude
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication

6.6.5.3 Extend: PRIVATE

La fonction <PrivateTDD> a pour but d'offrir des caractéristiques "privées" propres à chaque fabricant. La fonction <PrivateTDD> est un service complémentaire (voir le Tableau 35).

TABLEAU 35/T.611

Éléments de syntaxe minimaux de la fonction <PrivateTDD>

Élément de syntaxe	C	T	I/O	Codage en mode texte			Commentaire
				Mot clé	Paramètre	Valeur par défaut	
<PrivateFunction>	B	m	I	FUNCTION	"Private"	–	
<SubFunction>	B	m	I	SUBFUNC	<Subfunction-parameter>	–	Le choix des valeurs de ce paramètre appelle un complément d'étude
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Erreur renvoyée par l'application de communication

7 Méthode d'échange

Le présent article définit la manière dont les descriptions TDD et les données associées sont transférées entre les applications locales et les applications de communication.

Pour transférer les descriptions TDD et les données associées, la présente Recommandation définit une méthode d'échange abstraite entre les applications locales et les applications de communication. Cette méthode d'échange peut être réalisée par différents moyens, qui dépendent plus ou moins de la plate-forme.

Pour aider les concepteurs à écrire du code dont la portabilité soit bonne, les dépendances de la plate-forme ont été encapsulées à un niveau fonctionnel qui comprend les fonctions de la méthode d'échange de base (fonctions BEM).

La Figure 9 décrit cette situation.

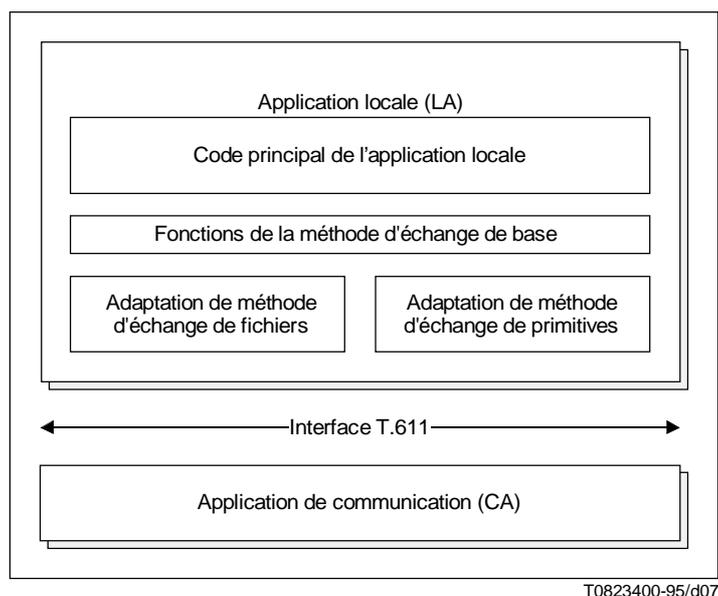


FIGURE 9/T.611

Utilisation des fonctions de la méthode d'échange de base

Les fonctions BEM présentent une interface fonctionnelle qui découple le code de l'application locale des méthodes d'échange réelles à utiliser.

Les méthodes d'échange réelles définies pour les diverses plates-formes sont de type à échange de fichiers ou à échange de primitives. Ces deux types de méthode d'échange sont entièrement détaillés au 7.3.

NOTE – Du fait que qu'il est requis uniquement que l'échange binaire de descriptions TDD et les données associées soient conformes à la présente Recommandation, l'utilisation des fonctions BEM dans le code de l'application locale est facultative. La seule prescription que doit remplir l'application locale est de se conformer à la méthode d'échange réelle définie pour chaque plate-forme. Aussi un concepteur d'application locale peut-il décider d'interfacer directement l'une des méthodes d'échange réelles définies.

7.1 Vue d'ensemble des fonctions de la méthode d'échange de base

NOTE – Les fonctions de la méthode d'échange de base décrites dans le présent paragraphe remplacent la méthode dont la documentation est donnée dans la version de la présente Recommandation datant de 1992. Pour le détail des modifications, se référer à l'Annexe F, dans laquelle la version antérieure des fonctions de la méthode d'échange de base a été incluse pour des raisons pratiques.

Différentes instances de la méthode d'échange de base sont possibles. Toutes ont en commun les fonctions de la méthode d'échange de base montrées dans le Tableau 36⁵⁾.

⁵⁾ Afin de différencier l'ensemble des fonctions BEM défini dans la présente Recommandation, de l'ensemble défini dans la version de 1992 de cette Recommandation, les noms des fonctions BEM commencent par la lettre majuscule "E".

TABLEAU 36/T.611

Résumé des fonctions de méthode d'échange

Fonction BEM	Objectif
ELogin	L'application locale ouvre un canal de communication entre l'application locale et l'application de communication
ELogout	L'application locale ferme le canal de communication ouvert par la fonction ELogin ()
EPutTDD	L'application locale envoie une description TDD à une application de communication
EPutData	L'application locale envoie des données se référant à une description TDD spécifique
EPollTDD	L'application locale interroge l'application de communication pour savoir si une description TDD de réponse est disponible
EGetTDD	L'application locale obtient une description TDD de réponse de la part de l'application de communication
EGetData	L'application locale obtient des données se référant à une description TDD spécifique
ESetAlarm	L'application locale place l'alarme dans l'application de communication. L'application de communication réveille ensuite l'application locale lors d'événements spécifiques
EAbortData	L'application locale abandonne le transfert de données
NOTE – La méthode d'échange de base a été conçue pour fonctionner dans de nombreuses configurations, telles celles où l'application locale et l'application de communication sont prises en charge par un même équipement ou celles où l'application de communication est un serveur de communication sur un réseau de zone locale (RZL). La seule hypothèse est que l'ICE soit accessible à partir de chaque application locale.	

En général les fonctions de la méthode d'échange de base se fondent sur une procédure d'ouverture de session (ELogin) qui rend disponible un identificateur Connection-ID. Cet identificateur est ensuite utilisé dans les appels ultérieurs des fonctions de la méthode d'échange de base. La procédure d'ouverture de session est comparable à l'ouverture d'un canal d'interaction entre une application locale et une application de communication.

7.1.1 Séquence des fonctions de la méthode d'échange de base

Premièrement, une application locale doit "ouvrir une session" (fonction ELogin) avec l'application de communication demandée. Aucun échange de descriptions TDD ne peut avoir lieu avant l'achèvement de la procédure d'ouverture de session. Par nature, cette procédure est un mécanisme synchrone, ce qui signifie que l'ouverture de session nécessite une réponse avant qu'une autre action puisse se produire.

Lorsqu'une application locale souhaite envoyer une description TDD de demande à une application de communication, elle doit passer par les étapes suivantes:

- constituer la description TDD de demande (par tout moyen approprié);
- appeler la fonction EPutTDD;
- appeler la fonction EPutData, s'il y a des données associées à transférer.

Lorsqu'une application locale veut être informée d'événements susceptibles de la concerner, elle doit:

- appeler la fonction EPollTDD.

Lorsqu'une application locale veut récupérer une description TDD de réponse (la fonction EPollTDD ci-dessus a indiqué que des descriptions TDD de réponse sont disponibles), elle doit:

- appeler la fonction EGetTDD;
- appeler la fonction EGetData, s'il y a des données associées à récupérer.

Lorsqu'une application locale reçoit une alarme de l'application de communication – applicable seulement si l'application de communication assure la fonction d'alarme comme indiqué dans l'ICE (voir l'article 9) et si l'application locale assure la fonction d'appel d'alarme – elle doit:

- appeler la fonction EPutTDD;
- appeler la fonction EGetTDD;
- appeler la fonction EGetData, s'il y a des données associées à récupérer.

Lorsqu'une application locale n'a plus besoin de dialoguer avec une application de communication, elle doit clore la session avec cette application de communication (fonction ELogout). L'application de communication sait ainsi que le trajet de communication LA-CA est interrompu et cesse donc d'utiliser le mécanisme d'alarme.

Les fonctions EPutTDD, EPolITDD, EGetTDD, ESetAlarm, ECallBackRoutine, ELogin et ELogout sont synchrones, c'est-à-dire que l'application locale ne peut poursuivre la procédure que lorsque ces fonctions ont renvoyé leur réponse. Les fonctions et leurs paramètres d'appel et de renvoi sont décrits au 7.2.

La Figure 10 illustre le comportement décrit plus haut.

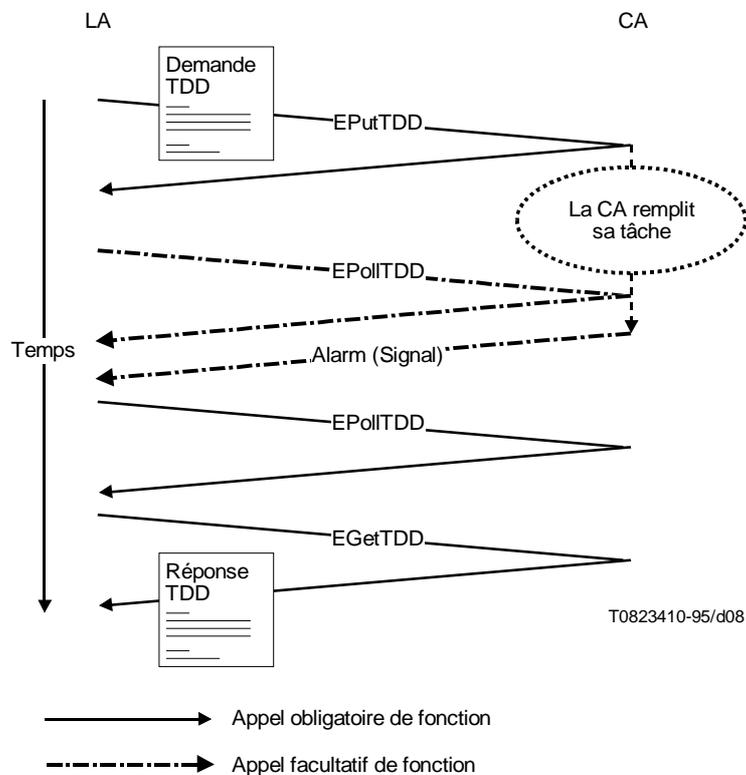


FIGURE 10/T.611

Exemple de séquence de fonctions de la méthode d'échange de base

On fait l'hypothèse que l'application locale a déjà ouvert une session dans l'application de communication. Les fonctions de transfert de données ne sont pas montrées.

7.1.2 Prise en charge des alarmes

Si l'application de communication assure la fonction d'alarme facultative, cela signifie qu'elle met en œuvre la fonction ESetAlarm. Cette fonction permet à une application de communication de "réveiller" une application locale donnée afin de lui envoyer les événements d'alarme.

La prise en charge de la fonction d'alarme doit être indiquée dans l'ICE (voir l'article 9).

7.1.3 Identificateur Connection-ID

Pour identifier un trajet entre application locale et application de communication, on définit l'identificateur Connection-ID. Cet identificateur est calculé par l'application de communication lors de l'appel de la demande d'ouverture de session. L'application locale doit utiliser cet identificateur pendant l'échange avec l'application de communication considérée jusqu'à la fermeture de la session par l'application locale.

NOTE – L'identificateur Connection-ID est différent de l'identificateur COM-ID: ce dernier identifie les événements de communication qui se produisent dans une application de communication.

7.1.4 Identificateur CA-ID

L'identificateur CA-ID identifie une application de communication. Cet identificateur permet à une application locale d'avoir des interactions simultanées avec plusieurs applications locales. L'identificateur CA-ID est donc un paramètre utilisé dans chaque appel de fonction EM.

7.2 Fonctions de la méthode d'échange de base

Les fonctions décrites ci-dessous remplacent ou complètent les fonctions du mécanisme d'échange de base de la version 1992 de la présente Recommandation (voir également l'Annexe F). Pour différencier les fonctions les unes des autres, les noms des fonctions définies dans la présente Recommandation commencent par la lettre majuscule "E".

Toutes les fonctions fonctionnent de manière synchrone, c'est-à-dire que l'appelant ne peut pas poursuivre l'exécution avant que la fonction appelée lui soit renvoyée.

Comme la mise en œuvre des fonctions BEM dépend de la plate-forme, ces fonctions sont décrites de manière générique dans les paragraphes qui suivent. La mise en œuvre des fonctions sur les diverses plates-formes est décrite dans la Partie IV de la présente Recommandation.

NOTE – L'ordre dans lequel les paramètres apparaissent dans la description suivante des fonctions BEM est important pour conserver la compatibilité.

Les types de données qui servent à la description générique des fonctions BEM sont définis dans le Tableau 37. Les types de données réelles auxquelles ils correspondent sont indiqués dans les articles appropriés de la Partie IV de la présente Recommandation.

Les classifications de direction "Input" et "Output" ont respectivement pour explications "paramètre devant être présenté à l'appel de fonction" et "paramètre valable au renvoi de la fonction".

TABLEAU 37/T.611

Types de données utilisés pour la description des fonctions BEM

Type de données	Explication
String	Chaîne de caractères
Integer	Nombre compris entre $+2^{15}$ et $(2^{15} - 1)$ au moins
Boolean	Variable qui peut prendre la valeur "vrai" ou "faux"
Memory address	Adresse à l'intérieur de la mémoire de l'application locale

7.2.1 Fonction ELogin (ouverture de session)

La fonction ELogin doit être assurée par l'application de communication. Elle est appelée par l'application locale avant tout échange de descriptions TDD de demande et de réponse entre l'application locale et l'application de communication.

7.2.1.1 Objet

La fonction ELogin renvoie à l'application locale un identificateur Connection-ID qui sera utilisé pendant toute la durée de l'interaction entre l'application locale et l'application de communication, jusqu'à la fermeture de la session par l'application locale.

En outre, la fonction ELogin permet de sélectionner une application de communication d'après des critères fournis par l'utilisateur et renvoie un identificateur CA-ID qui sert à réaliser l'adressage de l'application de communication sélectionnée dans les appels ultérieurs de fonctions BEM.

Pour la sélection d'une application de communication, on spécifie une chaîne contenant des mots clés tels que "FAX", "ECM" ou "EMAIL", séparés par des espaces, dans le paramètre Selector. Etant donné qu'il peut être actionné par des procédés externes, le mécanisme de sélection sort du cadre de la présente Recommandation.

Si le paramètre Selector est spécifié, le paramètre CA-ID ne doit pas l'être; au retour de l'appel de fonction, le paramètre CA-ID soit contient un identificateur d'application de communication valable pouvant être utilisé dans d'autres appels de fonction BEM, soit est vide si aucune application de communication n'était adaptée aux critères de sélection. Si le paramètre Selector n'est pas spécifié dans l'appel de fonction, il n'est pas nécessaire de procéder à une sélection. Dans ce cas, le paramètre CA-ID doit être spécifié.

La fonction ELogin est l'endroit où l'application de communication peut contrôler l'accès de l'application locale. Cela peut être réalisé en vérifiant les paramètres de nom d'ouverture de session (Login-name) et de mot de passe (Password) fournis par l'application locale. Cependant, le choix de l'étendue du contrôle des droits d'accès est laissé à l'instance d'application de communication.

7.2.1.2 Comportement

L'application de communication vérifie les paramètres d'appel de la fonction ELogin. S'ils correspondent, elle crée un identificateur Connection-ID que l'application locale devra utiliser par la suite dans les autres appels de fonctions BEM. De plus, si le sélecteur de paramètre spécifie les critères, la fonction renvoie l'identificateur CA-ID d'une application de communication qui correspond aux critères. Si l'identificateur Connection-ID renvoyé est mis à zéro, cela signifie que l'application de communication n'a pu se connecter à l'application locale.

7.2.1.3 Paramètres

Voir le Tableau 38.

TABLEAU 38/T.611

Paramètres de la fonction ELogin

Nom de paramètre	Type de données	Commentaire	Sens
Login-name (Nom d'ouverture de session)	String (chaîne)	Nom de l'utilisateur de l'application locale qui se connecte à l'application de communication (différent de l'identificateur LA-ID)	Paramètre d'entrée
Password (Mot de passe)	String (chaîne)	Chaîne comprenant le mot de passe	Paramètre d'entrée
Selector (Sélecteur)	String (chaîne)	Chaîne fournie par l'utilisateur qui spécifie les critères de connexion	Paramètre d'entrée
CA-ID (Identificateur d'application de communication)	Integer (entier)	Identificateur de l'application de communication qui accepte la connexion	Paramètre de sortie
Connection-ID (Identificateur de connexion)	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA. Renvoyé par l'application de communication si la fonction ELogin (ouverture de session) réussit. Dans le cas contraire (par exemple en cas d'échec d'identification), l'application de communication le met à zéro et indique le code d'erreur approprié dans le paramètre descripteur d'état (voir ci-dessous)	Paramètre de sortie
Status (Descripteur d'état)	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.2 Fonction EPutTDD

La fonction EPutTDD doit être assurée par l'application de communication. Elle peut être appelée par une application locale. La fonction EPutTDD prend également en charge l'identification des fichiers de données à transférer à l'application de communication.

De tels fichiers de données sont le plus souvent des fichiers de liste d'adresses ou des documents ou les deux.

7.2.2.1 Objet

La fonction EPutTDD transmet une description TDD de demande à une application de communication et permet à l'application locale de spécifier les fichiers qu'il convient de transférer à l'application de communication. Le transfert effectif des fichiers de données à la CA est réalisé grâce à la fonction EPutData.

7.2.2.2 Comportement

L'application de communication copie la description TDD de demande transmise par la fonction EPutTDD dans ses structures internes. Le résultat est communiqué immédiatement à l'application locale ayant adressé la demande. Si l'application locale demande un identificateur Data-ID en spécifiant une adresse d'identificateur Data-ID, l'application de communication renvoie l'identificateur d'un groupe de données, en tant qu'identificateur Data-ID.

Si cet identificateur a la valeur NULL, aucun fichier de données (ou mémoire tampon) ne doit être transmis à l'application de communication. Si l'identificateur Data-ID n'a pas la valeur NULL, l'application locale accomplit les transferts de fichiers de données ultérieurs avec la fonction EPutData. Ces transferts doivent être effectués immédiatement après l'appel de la fonction EPutTDD.

NOTE – Le paramètre Data-ID peut être valable pendant un temps limité seulement.

7.2.2.3 Paramètres

Voir le Tableau 39.

TABLEAU 39/T.611

Paramètres de la fonction EPutTDD

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID (Identificateur d'application de communication)	Integer (entier)	Identificateur de l'application de communication tel que renvoyé par la fonction ELogin	Paramètre d'entrée
TDD-location (Emplacement des TDD)	Memory address (adresse mémoire)	Emplacement mémoire des TDD à passer à l'application de communication. Après retour de la fonction les descriptions TDD de l'application locale peuvent être effacées ou servir à d'autres fins	Paramètre d'entrée
TDD-size (Taille des TDD)	Integer (entier)	Taille des descriptions TDD passées à l'application de communication, en octets	Paramètre d'entrée
Data-ID (Identificateur de données)	Integer (entier)	Identificateur du groupe de données, renvoyé par l'application de communication sur demande	Paramètre d'entrée/ sortie
Status (Descripteur d'état)	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.3 Fonction EPutData

Cette fonction permet le transfert effectif de données (listes d'adresses et/ou fichiers de données) de l'application locale à l'application de communication. Suivant l'instance, les données peuvent être des fichiers ou des mémoires tampons.

7.2.3.1 Objet

Le groupe de fichiers transférés dans un appel de fonction est relié à une description TDD de demande antérieure qui a été transmise à l'application de communication par l'intermédiaire d'un appel de fonction EPutTDD antérieur.

Ce groupe est identifié par l'identificateur Data-ID, que l'application de communication a renvoyé comme réponse à l'appel précédent de fonction EPutTDD.

7.2.3.2 Comportement

Cette fonction est appelée par l'application locale après que la description TDD de demande correspondante a été transmise à l'application de communication.

L'application locale construit une structure de données (Data-Descriptor) qui décrit où sont situés les fichiers/mémoires tampons à transmettre à l'application de communication. La spécification de la structure de données dépend du langage mais la compatibilité mutuelle des plates-formes garantit que la structure binaire ne dépend pas des plates-formes ou des langages.

L'application locale peut transférer toutes les données d'un coup ou en de multiples appels. Le paramètre *Next* (suivant) indique si l'appel de fonction est le dernier du groupe de données ou si d'autres appels de fonctions correspondant à ce groupe de données suivent.

7.2.3.3 Paramètres

Voir le Tableau 40.

TABLEAU 40/T.611

Paramètres de la fonction EPutData

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA	Paramètre d'entrée
CA-ID (Identificateur d'application de communication)	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
Data-ID (Identificateur de données)	Integer (entier)	Identificateur du groupe de données, renvoyé par l'application de communication	Paramètre d'entrée
Data-Descriptor (Descripteur de données)	Memory address (adresse mémoire)	Adresse mémoire d'une structure de données spécifiant si les fichiers/mémoires tampons doivent être transférés à l'application de communication	Paramètre d'entrée
Next (Suivant)	Boolean (variable booléenne)	Indicateur établissant si davantage de données seront transportées dans un appel ultérieur à la fonction EPutData ou si aucune autre donnée ne va suivre (fin du groupe de données)	Paramètre d'entrée
Status (Descripteur d'état)	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.4 Fonction EPollTDD

La fonction EPollTDD interroge l'application de communication pour savoir combien de descriptions TDD de réponse attendent d'être traitées par l'application locale à l'origine de la demande. La fonction EPollTDD renvoie le nombre de descriptions TDD de réponse en attente ainsi que le type et la taille de la première réponse de descriptions TDD de réponse qui sera renvoyée par le prochain appel à la fonction EGetTDD.

En outre, la fonction EPollTDD permet de sélectionner le type de description TDD qui doit être renvoyé par l'application de communication.

7.2.4.1 Objet

La fonction EPollTDD permet d'interroger l'application de communication pour savoir le nombre de descriptions TDD de réponse qui attendent d'être récupérées.

7.2.4.2 Comportement

Cette fonction permet à l'application locale de sélectionner le type de descriptions TDD de réponse à rechercher en donnant un paramètre spécifique de type de TDD. Si l'application locale ne spécifie pas de type de description TDD (indicateur TDD-type mis à zéro), la fonction choisit, si disponible, n'importe quel type de description TDD et renvoie le type sélectionné. La description TDD sélectionnée pourra, s'il y a lieu, être récupérée par un appel de fonction EGetTDD ultérieur.

La fonction EPollTDD renvoie également la taille de la description TDD de réponse qui sera renvoyée, s'il y a lieu. L'application de communication devrait alors affecter une capacité de mémoire suffisante pour recevoir la description TDD de réponse renvoyée par le prochain appel à la fonction EGetTDD. La taille donnée par l'application de communication constitue une indication de la taille minimale de la mémoire tampon qui contiendra la description TDD de réponse.

En outre, le nombre de descriptions TDD de réponse en attente – du type sélectionné – est renvoyé. Si le compte est à zéro, cela signifie qu'il n'y a aucune description TDD de réponse – du type sélectionné – en attente.

Lorsque l'application de communication dispose de nombreuses descriptions TDD de réponse, elle choisit celle qu'elle renverra en premier. Cette description TDD de réponse sera transmise à l'application locale lors du prochain appel de la fonction EGetTDD émis par cette même application locale.

Lorsque aucune description TDD de réponse n'est disponible pour l'application locale à l'origine de la demande, le décompte de descriptions TDD (TDD-Count) est mis à zéro, auquel cas la taille de description TDD (TDD-Size) renvoyée est également mise à zéro.

Si la description TDD soumise est erronée ou inconnue, à savoir si l'en-tête <TDDHeader> manque, la fonction peut renvoyer la description TDD de demande originale. Dans ce cas, l'application de communication doit mettre le paramètre TDD-type renvoyé à zéro et le descripteur d'état renvoyé à une valeur non nulle.

Les valeurs du paramètre TDD-type sont montrées dans le Tableau 41.

TABLEAU 41/T.611

Attribution des types de descriptions TDD

Description TDD de réponse	TDD-type	Description TDD de réponse	TDD-type
Pas de TDD-type ^{a)}	00 _{HEX}	Réponse DISPATCH	35 _{HEX}
Réponse SENDACK	10 _{HEX}	Réponse PREVIEW	36 _{HEX}
Réponse RECEIVE	20 _{HEX}	Réponse PRINT	40 _{HEX}
Réponse COPY	30 _{HEX}	Réponse CONVERT	41 _{HEX}
Réponse DELETE	31 _{HEX}	Réponse CHECK	42 _{HEX}
Réponse CANCEL	32 _{HEX}	Réponse EXTEND	50 _{HEX}
Réponse PURGE	33 _{HEX}	Réponse NATIONAL	60 _{HEX}
Réponse RESCHEDULE	34 _{HEX}	Réponse PRIVATE	70 _{HEX}

^{a)} Utilisé comme sélecteur (en entrée), le caractère 00_{HEX} est mis pour "tous types de description TDD"; lorsque cette valeur est renvoyée, cela signifie qu'une description TDD erronée ou inconnue a été soumise ou qu'aucune description TDD de réponse n'est disponible.

7.2.4.3 Paramètres

Voir le Tableau 42.

TABLEAU 42/T.611

Paramètres de la fonction EPollTDD

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID (Identificateur d'application de communication)	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
TDD-type (Type de TDD)	Integer (entier)	Sélectionne/renvoie le type de descriptions TDD de réponse attendu/disponible	Paramètre d'entrée/ sortie
TDD-size (Taille des descriptions TDD)	Integer (entier)	Taille de la prochaine description TDD de réponse à récupérer	Paramètre de sortie
TDD-count (Décompte de descriptions TDD)	Integer (entier)	Nombre de descriptions TDD de réponse en attente. La valeur zéro indique qu'aucune réponse TDD n'est en attente	Paramètre de sortie
Status (Descripteur d'état)	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.5 Fonction EGetTDD

La fonction EGetTDD doit être assurée par l'application de communication. Elle peut être appelée par une application locale. Elle permet de récupérer une description TDD de réponse et d'obtenir un identificateur de groupe de données pour récupérer les fichiers de données/les mémoires tampons qui les accompagnent.

7.2.5.1 Objet

La fonction EGetTDD permet à l'application locale de récupérer une description TDD de réponse en mémoire et d'obtenir un pointeur vers un groupe de fichiers de données/de mémoires tampons. Le pointeur est utile à tous les types de description TDD qui agissent avec des fichiers de données/des mémoires tampons, par exemple les fonctions Receive ou Trace:COPY.

7.2.5.2 Comportement

L'application locale indique l'adresse mémoire où l'application de communication doit copier une description TDD de réponse disponible pour l'application locale.

L'application de communication doit renvoyer à l'application locale la description TDD de réponse qui a été désignée par la précédente fonction EPollTDD émise par cette même application locale. L'application locale doit avoir préparé une zone de description TDD de réponse destinataire dans ses structures internes. L'appel d'une fonction EGetTDD par une application locale doit toujours être précédé d'un appel de fonction EPollTDD. En cours d'appel, l'application locale doit prérégler le paramètre taille des descriptions TDD à la taille de sa zone de réponse. Le paramètre taille des descriptions TDD renvoyé doit conserver la taille de la description TDD de réponse que l'application de communication lui a donnée.

Si l'application locale effectue deux ou plusieurs appels consécutifs à la fonction EGetTDD (sans appel intermédiaire à la fonction EPollTDD), l'application de communication peut renvoyer les informations non valables.

La fonction EGetTDD renvoie également l'identificateur Data-ID sous forme d'un pointeur pour recevoir les fichiers de données associées (avec un appel de fonction ultérieur EGetTDD).

Si la valeur de paramètre Data-ID n'est pas nulle, alors les fichiers de données doivent être récupérés par un appel ultérieur à la fonction EGetData. Si cette valeur est nulle, cela signifie qu'aucune donnée supplémentaire n'est à récupérer de l'application de communication.

7.2.5.3 Paramètres

Voir le Tableau 43.

TABLEAU 43/T.611

Paramètres de la fonction EGetTDD

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID (Identificateur d'application de communication)	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
TDD-location (Emplacement des TDD)	Memory address (adresse mémoire)	Indique l'emplacement où enregistrer la description TDD de réponse	Paramètre d'entrée
TDD-size (Taille des TDD)	Integer (entier)	Taille de la mémoire tampon réservée pour enregistrer la description TDD de réponse	Paramètre d'entrée/ sortie
Data-ID (Identificateur de données)	Integer (entier)	Identificateur du groupe de données, s'il y a lieu	Paramètre de sortie
Status (Descripteur d'état)	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.6 Fonction EGetData

Cette fonction permet d'assurer le transfert de fichiers de données (ou de mémoires tampons) de l'application de communication vers l'application locale. Elle est appelée par l'application locale.

7.2.6.1 Objet

Cette fonction est utilisée pour récupérer les données de l'application de communication après réception d'une description TDD de réponse (avec la fonction EGetTDD) qui spécifie qu'il y a des données à récupérer (le paramètre Data-ID n'est pas zéro).

7.2.6.2 Comportement

Immédiatement après la fonction EGetTDD, l'application locale appelle la fonction EGetData afin de recevoir les données correspondantes (fichiers ou mémoires tampons). L'application de communication renvoie un descripteur de données contenant l'information sur les données reçues. L'application de communication précise également si l'application locale doit effectuer des appels supplémentaires à la fonction EGetTDD pour récupérer d'autres données associées.

Le groupe de fichiers transférés dans un appel de fonction est lié à une seule description TDD de réponse (par l'intermédiaire d'un appel de fonction EGetTDD). Ce groupe est identifié par le paramètre Data-ID (fourni par l'application locale).

7.2.6.3 Paramètres

Voir le Tableau 44.

TABLEAU 44/T.611

Paramètres de la fonction EGetData

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID (Identificateur d'application de communication)	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
Data-ID (Identificateur de données)	Integer (entier)	Identificateur du groupe de données	Paramètre d'entrée
Data-Descriptor (Descripteur de données)	Memory address (adresse mémoire)	Adresse d'enregistrement d'un descripteur pour les données à récupérer, transmise à l'application de communication en entrée et complétée par l'application de communication en sortie	Paramètre d'entrée/ sortie
Next (Suivant)	Boolean (variable booléenne)	Indicateur établissant si d'autres données doivent être récupérées	Paramètre de sortie
Status (Descripteur d'état)	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.7 Fonction ESetAlarm

La fonction ESetAlarm est une option qui peut être assurée par l'application de communication, auquel cas elle peut éventuellement être appelée par une application locale. Si la fonction ESetAlarm est utilisée par une application locale, cette application locale doit assurer la fonction AlarmHandler. Une fonction AlarmHandler reçoit les alarmes ultérieures créées par l'application de communication.

L'application de communication doit déclarer la prise en charge de la fonction ESetAlarm dans l'ICE.

7.2.7.1 Objet

Cette fonction permet à l'application locale d'enregistrer auprès de l'application de communication connectée le type d'événements d'alarme que l'application locale prend en charge. Cette fonction n'est pas appelée si l'application de communication ne comprend pas le mécanisme d'alarme. L'information précisant si l'application de communication comprend le mécanisme d'alarme peut être obtenue de l'élément ALARM de l'ICE (voir l'article 9).

La fonction ESetAlarm indique à l'application de communication qu'elle peut donner l'alarme à l'application locale en appelant la fonction AlarmHandler qui lui est fournie.

L'application locale peut désactiver temporairement un événement d'alarme en invalidant l'adresse correspondante de la fonction AlarmHandler, c'est-à-dire en mettant cette adresse à zéro. Le rétablissement de l'événement d'alarme pour ce type d'alarme est effectué en autorisant l'adresse de la fonction AlarmHandler, c'est-à-dire en mettant l'adresse au point d'entrée effectif de la fonction AlarmHandler.

Les événements d'alarme sont définis dans le Tableau 45.

7.2.7.2 Comportement

L'application de communication qui prend en charge la fonction ESetAlarm doit enregistrer l'emplacement de la fonction AlarmHandler affecté par l'application locale. Au minimum, l'application de communication peut enregistrer autant d'emplacements de fonctions AlarmHandler qu'il y a d'applications locales en session ouverte avec elle. Si un événement enregistré dans une application locale particulière se produit, l'application de communication peut alors éveiller une application locale particulière en appelant sa fonction AlarmHandler.

TABLEAU 45/T.611

Evénements d'alarme de la fonction ESetAlarm

Nom de l'événement	Commentaire
ASYNC_RESPONSES	L'application locale n'interrogera pas l'application de communication; l'application de communication devrait envoyer une alarme chaque fois qu'une description TDD est disponible
QUEUE_FULL	L'application de communication n'acceptera plus de descriptions TDD ultérieures sauf si l'application de communication est interrogée
DOCUMENT_RECEIVED	Un nouveau document peut être reçu
CONNECTION_LOST	La connexion LA-CA est perdue
SEND_SUCCESS	L'envoi d'un document a réussi
SEND_FAILED	L'envoi d'un document a échoué
CORRUPTED_TDD	L'application de communication a reçu une description TDD corrompue/non reconnaissable
SEND_EVENT	Un événement d'envoi s'est produit dans l'application de communication
RECEIVE_EVENT	Un événement de réception s'est produit dans l'application de communication
CA_WILL_STOP	L'application de communication ne traitera plus les demandes. Terminer la session immédiatement
ALARMS_UNAVAILABLE	Les alarmes ne sont plus disponibles
TDD_RESP_AVAILABLE	Une description TDD de réponse est disponible. L'application locale devrait interroger l'application de communication le plus tôt possible
NOTE – Certaines alarmes nécessitent des données supplémentaires pour fournir des informations précises sur l'événement.	

7.2.7.3 Paramètres

Voir le Tableau 46.

TABLEAU 46/T.611

Evénement d'alarme de la fonction ESetAlarm

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
Alarm-event (Evénement d'alarme)	Integer (entier)	Combinaison des types d'alarme auxquels l'application locale pourrait réagir	Paramètre d'entrée
AlarmHandler	Memory address (adresse mémoire)	Information donnant l'adresse du point d'entrée de la fonction AlarmHandler. Ce point d'entrée doit être appelé par l'application de communication lorsque l'un des événements enregistrés se produit Ce paramètre (ainsi que la fonction AlarmHandler elle-même) dépend de la plate-forme	Paramètre d'entrée
Status	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.7.4 Fonction AlarmHandler

Selon la plate-forme, l'alarme déclenchée peut avoir des paramètres qui détaillent la raison de l'alarme.

La fonction AlarmHandler définit un mécanisme qui permet à une application de communication d'avertir l'application locale que certaines descriptions TDD de réponse sont disponibles. L'utilisation de ce mécanisme facultatif peut améliorer la commande de flux entre applications locales et applications de communication dans les systèmes lourdement chargés.

La fonction AlarmHandler est mise en œuvre par l'application locale. L'application de communication appelle cette fonction lorsqu'elle a besoin de déclencher une alarme. Les alarmes que l'application de communication peut déclencher sont spécifiées par l'application locale au moyen de la fonction ESetAlarm.

Certaines alarmes peuvent comporter des paramètres. Ceux-ci sont communiqués à l'application locale avec l'appel de fonction.

Certaines alarmes supposent que l'application locale interroge l'application de communication après réception de l'alarme (par exemple, lorsque l'application de communication a reçu un nouveau document). Si l'application locale n'interroge pas l'application de communication assez rapidement, celle-ci peut déclencher l'alarme de façon répétée jusqu'à ce que l'application locale exécute l'action attendue.

Etant donné que la fonction AlarmHandler dépend de la plate-forme, elle est décrite plus en détail dans les articles appropriés de la Partie IV de la présente Recommandation.

7.2.8 Fonction EAbortData

La fonction EAbortData permet d'annuler un transfert de données en cours. Elle a également pour effet d'annuler les descriptions TDD correspondantes.

7.2.8.1 Objet

La fonction est appelée par l'application locale afin d'annuler un transfert de données lancé par une fonction EPutData ou EGetData.

7.2.8.2 Comportement

Au moment d'annuler un transfert de données par la fonction EPutData, l'application de communication annule la description TDD de demande correspondante qui a été transmise par l'appel de fonction EPutTDD antérieur.

Au moment d'annuler un transfert de données par la fonction EGetData, l'application de communication conserve la description TDD de réponse correspondante qui a été récupérée par l'appel de fonction EGetTDD antérieur.

L'annulation des transferts de données dans l'un ou l'autre sens ne peut avoir lieu qu'entre deux appels de fonctions EGetData ou EPutData.

Etant donné que la responsabilité du transfert de données entre l'application locale et l'application de communication incombe à l'application locale, l'appel de la fonction EAbortData après le dernier bloc de données est sans effet (le transfert de données étant de toute façon achevé). Aussi l'appel de la fonction EAbortData avant le premier bloc de données a-t-il pour effet d'annuler dans son intégralité le transfert de données et la description TDD correspondante.

7.2.8.3 Paramètres

Voir le Tableau 47.

7.2.9 Fonction ELogout

La fonction ELogout doit être assurée par l'application de communication. Elle est appelée par l'application locale à la fin de tout échange LA-CA de descriptions TDD de demande et de réponse.

7.2.9.1 Objet

La fonction ELogout renvoie à l'application locale un descripteur d'état qui indique si l'interaction LA-CA a pris fin correctement.

7.2.9.2 Comportement

Avant de mettre fin au dialogue LA-CA, l'application de communication peut (mais n'est pas tenue de le faire) traiter toutes les descriptions TDD de demande en suspens qui ont été émises par cette application locale.

TABLEAU 47/T.611

Paramètres de la fonction EAbortData

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
Data-ID	Integer (entier)	Identificateur du groupe de données	Paramètre d'entrée
Status	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.2.9.3 Paramètres

Voir le Tableau 48.

TABLEAU 48/T.611

Paramètres de la fonction ELogout

Nom de paramètre	Type de données	Commentaire	Sens
Connection-ID	Integer (entier)	Identificateur (pointeur) du canal de données LA-CA renvoyé par la fonction ELogin (ouverture de session)	Paramètre d'entrée
CA-ID	Integer (entier)	Identificateur de l'application de communication	Paramètre d'entrée
Status	Integer (entier)	Code renvoyé; la valeur nulle indique la réussite de l'opération	Paramètre de sortie

7.3 Mise en œuvre des fonctions de la méthode d'échange de base

Les fonctions de méthode d'échange de base offrent une interface fonctionnelle qui doit être adaptée à la plate-forme sous-jacente d'une manière bien définie pour fournir une interface binaire compatible.

A cette fin, la présente Recommandation définit deux mécanismes d'échange réels:

- 1) la méthode d'échange de primitives; et
- 2) la méthode d'échange de fichiers.

La méthode d'échange de primitives est rapide mais est dépendante de la plate-forme; la méthode d'échange de fichiers est beaucoup moins dépendante de la plate-forme mais est plus lente et ne permet pas de mettre en œuvre toutes les caractéristiques offertes par les fonctions de la méthode d'échange de base.

Les plates-formes peuvent imposer le type de méthode d'échange à mettre en œuvre. Voir la Partie IV de la présente Recommandation.

7.3.1 Méthode d'échange de primitives

La méthode d'échange de primitives utilise des appels de fonctions. Elle requiert donc de la part de l'application de communication de fournir un point d'entrée pour chacune des fonctions de la méthode d'échange de base. Pour réaliser la compatibilité binaire, ce point d'entrée doit être fourni de manière conforme aux définitions que donne la présente Recommandation pour chaque plate-forme (voir la Partie IV de la présente Recommandation).

7.3.2 Méthode d'échange de fichiers

La méthode d'échange de fichiers requiert que les descriptions TDD soient échangées par l'intermédiaire de fichiers. Aussi l'application de communication et l'application locale doivent-elles se mettre d'accord sur les répertoires de fichiers communs et les utiliser en partage. L'application de communication doit donc déclarer les chemins complets réservés à ces répertoires dans la partie appropriée de l'ICE.

On doit configurer les trois répertoires de fichiers, ou en termes plus abstraits, trois zones ci-après:

- une zone d'entrée pour les descriptions TDD qui sont transférées de l'application locale à l'application de communication;
- une zone pour les descriptions TDD de réponse;
- une zone pour les tâches qui n'ont pas pu être traitées en raison d'erreurs syntaxiques ou autres. L'application de communication supprime les descriptions TDD erronées de la zone d'entrée et les place dans cette zone.

Les symboles COM_JOB, COM_ACK et COM_ERR font respectivement référence à ces trois zones.

Comme ils sont propres à l'instance et à l'installation, les chemins complets vers ces zones sont configurables pendant l'installation de l'application de communication.

Si une application de communication doit prendre en charge des connexions à des applications de communication multiples, on suggère que l'application de communication établisse trois zones "mères" (répertoires) comme décrit plus haut et crée une sous-zone (sous-répertoire) dans chaque zone "mère" pour chaque application locale qui peut être connectée. Le nom de chaque sous-zone peut être choisi à partir de l'identificateur LA-ID des applications locales connectantes.

L'adaptation des fonctions de la méthode d'échange de base est relativement complexe, du fait que ces fonctions doivent être émulées par l'adaptateur.

NOTE – Une instance d'application locale peut – bien que la présente Recommandation ne l'encourage pas – également s'interfacer directement avec la méthode d'échange de fichiers sans utiliser les fonctions de la méthode d'échange de base. Le problème de cette approche est une plus grande difficulté pour obtenir la portabilité vers d'autres plates-formes utilisant des primitives.

7.3.2.1 Transfert de descriptions TDD

Pour émettre une demande de fonction, l'application locale place une description TDD sous forme d'un fichier dans la zone COM_JOB.

Certaines instances, principalement des plates-formes monotâche, requièrent également que l'application locale exécute un programme SYNC. Ce programme SYNC (synchronisation), garantit que l'application de communication reçoit le contrôle et peut traiter les descriptions TDD. Le descripteur de CA (CA-Descriptor) de l'application de communication correspondante établit s'il est nécessaire d'exécuter un programme SYNC.

Lorsque l'application de communication a fini de traiter la description TDD, elle place la description TDD de réponse dans la zone de réponse COM_ACK et efface la description TDD de demande de la zone d'entrée COM_JOB. La description TDD de réponse placée par l'application de communication prend le même nom que la description TDD de demande antérieure.

Le cycle complet du transfert de tâche est montré dans la Figure 11.

NOTE – Il n'est pas garanti que les descriptions TDD de réponse seront fournies dans la même séquence que les descriptions TDD de demande. Le traitement n'est pas obligatoirement séquentiel.

7.3.2.2 Traitement des erreurs

Les descriptions TDD de demande qui ne peuvent être traitées par l'application de communication (par suite d'erreurs de syntaxe, etc.) sont copiées dans la zone COM_ERR et supprimées de la zone COM_JOB. Le mot clé FATAL est inséré à la fin de l'enveloppe et un message d'erreur de la forme [Num/Text] est placé dans le champ de paramètre associé.

7.3.2.3 Mécanisme SYNC

Si une exécution de programme SYNC est requise par l'application de communication, l'application locale doit procéder de la manière suivante:

- après qu'une description TDD de demande eut été enregistrée, l'application locale doit envoyer un mécanisme SYNC à l'application de communication (fonction "SyncJob");
- avant que l'application locale ait interrogé la description TDD de réponse, un autre mécanisme SYNC doit être envoyé (fonction "SyncAck").

Sous les systèmes d'exploitation DOS, OS/2, UNIX et de type UNIX, le mécanisme SYNC doit être mis en œuvre par exécution du programme "APPLI/COM" de l'application locale.

Les fonctions "SyncJob" et "SyncAck" (voir ci-dessus) sont codées de la manière suivante:

SyncJob: EXEC APPLI/COM JOB

SyncAck: EXEC APPLI/COM ACK

EXEC désigne la commande propre à la plate-forme qui sert à lancer l'exécution d'un programme exécutable.

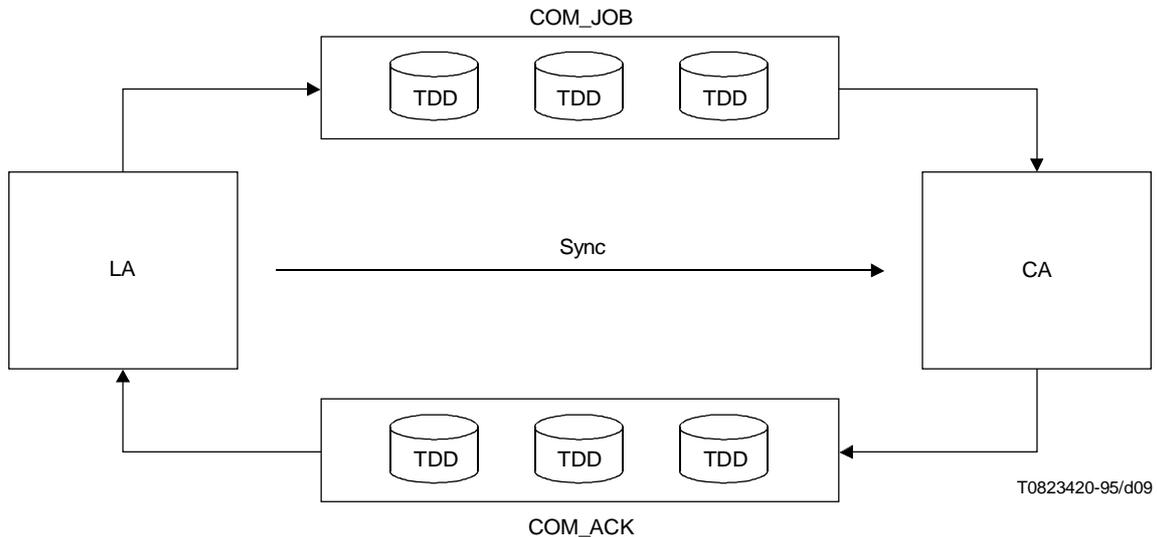


FIGURE 11/T.611

Méthode d'échange de fichiers – Transfert de descriptions TDD

8 Formats de transfert

Pour un fonctionnement correct, certaines conditions s'appliquent au format des fichiers sortants et entrants échangés entre application locale et application de communication. Les fichiers entrants et sortants ont un format particulier qui n'est pas nécessairement le même que celui qu'utilisent les fichiers d'émission, par exemple un fichier de traitement de texte à échanger par le mode de base du service télécopie. Dans ce cas, des conversions de formats sont nécessaires (qui doivent être traitées par l'application de communication).

Pour l'échange de documents entre application locale et application de communication, plusieurs formats de transfert sont définis par la présente Recommandation. Ces formats de transfert s'appliquent aux fichiers de transfert tels que définis au 2.1. Il ne faut pas confondre ces formats avec le format transmis à travers le réseau (fichiers d'émission) par les services de télécommunication ni avec le format utilisé par les descriptions TDD (codage des descriptions TDD).

Plusieurs formats de transfert sont possibles:

- formats de transfert en mode texte;
- formats de transfert en mode graphique vectoriel;
- formats de transfert en mode transparent;
- formats de transfert privés, en mode texte ou graphique.

La présente Recommandation définit:

- 3 formats de transfert en mode texte (ASCII étendu APPLI/COM, ASCII standard APPLI/COM et format télétext);
- 1 format de transfert en mode graphique (APPLI/COM TIFF);
- le format de transfert transparent.

Le format transparent ne peut être utilisé que si le document doit être émis comme fichier binaire par le service de télécommunication. Dans ce cas, aucune conversion du contenu du fichier n'a lieu et le fichier doit donc être transmis tel quel.

La présente Recommandation permet d'utiliser des formats de transfert privés. D'autres formats de transfert (par exemple en langage "PostScript") peuvent donc être mis en œuvre par les fabricants d'applications de communication pour adapter les formats locaux des programmes d'application généralement utilisés (par exemple, traitement de texte, bases de données ou tableurs). Pour informer l'application locale de cette prise en charge, l'application de communication doit placer le mot clé ADDCONV dans le descripteur de CA de l'ICE (voir l'article 9).

Une application de communication prenant en charge un format de transfert supplémentaire doit s'assurer que les documents sont traités correctement pour tous les services pris en charge par l'application de communication. Les restrictions imposées par le service lui-même doivent être traitées de manière transparente. Quoi qu'il en soit, les formats de transfert pris en charge par les fabricants d'applications de communication doivent être décrits dans leur documentation correspondante.

Le Tableau 49 indique comment les formats de transfert doivent être pris en charge par l'application de communication selon le système d'exploitation et le service de télécommunication qu'elle offre.

TABLEAU 49/T.611

Liste des formats de transfert

Formats de transfert	Convert-id	Signifie
ASCII étendu APPLI/COM	ASCII437	Format de transfert ASCII tel que défini au 8.1.1. L'utilisation de ce format est limitée à certains services. Voir 8.4
	ASCII	Format de transfert ASCII tel que défini pour l'application de communication qui le prend en charge. Ce format de transfert peut différer du format de transfert ASCII437 sur les systèmes qui ne prennent pas en charge le même jeu de caractères ASCII étendu
ASCII standard APPLI/COM	T.50	Format de transfert de la Recommandation T.50 (IRA) tel que défini dans 8.1.2. L'utilisation de ce format est limitée à certains services. Voir 8.4
Format télétext	T.61	Format de transfert de la Recommandation T.61 tel que défini au 8.2. L'utilisation de ce format est limitée à certains services. Voir 8.4.
APPLI/COM TIFF	TIFF	Format de transfert TIFF tel que défini dans 8.3. L'utilisation de ce format est limitée à certains services. Voir 8.4.
Format transparent	VOID	Signifie "aucune conversion à effectuer". L'utilisation est limitée au transfert transparent de documents dans divers modes de "transfert de fichiers" des services de télécommunication

Les formats de transfert définis dans la présente Recommandation peuvent être lus et créés pour les services appropriés, sous les systèmes d'exploitation appropriés, par les applications de communication conformes.

NOTE – Les formats de transfert en mode texte pour le service télécopie sont pris en charge dans le sens sortant uniquement.

Les documents qui sont transférés à l'application de communication dans un format en mode texte doivent être édités par l'application de manière que le format et le jeu de caractères correspondent aux prescriptions du service, à savoir le "nombre de caractères par ligne", le "nombre de lignes par page" et "l'interligne", ainsi que les attributs comme "soulignement", "exposant" et "indice".

L'application de communication peut rejeter les documents dont le format ou le jeu de caractères est incorrect.

NOTE – Si une application locale souhaite réaliser l'indépendance du service, cette application peut utiliser l'un des formats de type "ASCII" pour le transfert des documents, car ils représentent les seuls formats de transfert qui recouvrent la plupart des services.

8.1 Formats de transfert APPLI/COM: ASCII étendu ou normalisé

Les codes de format montrés dans le Tableau 50 sont définis pour les formats de transfert ASCII étendu APPLI/COM et ASCII normalisé APPLI/COM. A noter que tous les services ne permettent pas toutes les spécifications de format énumérées (voir 8.4). Une application de communication est autorisée à ne pas tenir compte des séquences d'échappement. En outre, selon les polices de caractères résidentes utilisées pour effectuer les conversions d'ASCII à T.4 (ou T.6), l'application de communication est autorisée à changer de ligne ou de page lorsque nécessaire, sauf si cela est expressément interdit par l'application locale; dans ce cas, l'application de communication peut rejeter la conversion ou l'effectuer à un niveau de qualité inférieur.

TABLEAU 50/T.611

Opérateurs de formatage pour les formats de transfert ASCII APPLI/COM

Format	Valeurs possibles	HEX	ASCII	Valeur par défaut
Orientation	Portrait	1B 4F 30	ESC O 0	√
	Italien	1B 4F 31	ESC O 1	
Corps	Pica 10	1B 50 30	ESC P 0	√
	Pica 12	1B 50 31	ESC P 1	
	Pica 15	1B 50 32	ESC P 2	
Interligne	6 lignes/pouce (1 interligne)	1B 4C 30	ESC L 0	√
	4 lignes/pouce (1,5 interligne)	1B 4C 31	ESC L 1	
	3 lignes/pouce (2 interlignes)	1B 4C 32	ESC L 2	
	12 lignes/pouce (2,5 interlignes)	1B 4C 33	ESC L 3	
Attributs	Soulignement hors fonction	1B 55 30	ESC U 0	√
	Soulignement en fonction	1B 55 31	ESC U 1	
	Exposant hors fonction	1B 41 30	ESC A 0	√
	Exposant en fonction	1B 41 31	ESC A 1	
	Indice hors fonction	1B 56 30	ESC V 0	√
	Indice en fonction	1B 56 31	ESC V 1	
	Caractères gras hors fonction	1B 42 30	ESC B 0	√
	Caractères gras en fonction	1B 42 31	ESC B 1	
	Biffage hors fonction	1B 53 30	ESC S 0	√
	Biffage en fonction	1B 53 31	ESC S 1	
	Italiques hors fonction	1B 49 30	ESC I 0	√
Italiques en fonction	1B 49 31	ESC I 1		
Mise en page	Retour à la ligne non autorisé	1B 54 30	ESC T 0	
	Retour à la ligne autorisé	1B 54 31	ESC T 1	√
	Changement d'orientation de page non autorisé	1B 52 30	ESC R 0	
	Changement d'orientation de page autorisé	1B 52 31	ESC R 1	√
	Ligne nouvelle	0D 0A	CR LF	
	Page nouvelle	0D 0C	CR FF	
Sélection de police	Sélection du numéro de police n	1B 43 'n'	ESC C 'n'	'n' = 0

NOTE – Les polices à sélectionner doivent être déclarées dans l'ICE (voir l'article 9). Les polices sont normalement de corps et d'espacement de caractères fixes. Le processus de conversion doit assurer le non-recouvrement des caractères ainsi que l'absence de pertes dans le processus. Pour les grands corps de caractères grands, l'interligne peut augmenter et le formatage changer. Cependant, l'application de communication doit s'assurer qu'aucun contenu ne soit perdu et que les lignes de texte soient lisibles.

8.1.1 Jeu de caractères du format de transfert ASCII étendu APPLI/COM

Les caractères utilisés par le format de transfert ASCII étendu APPLI/COM pour les services télétexte, télex et télécopie sont indiqués dans les Tableaux 51, 52 et 53⁶⁾. Ce format de transfert peut être appelé de deux manières:

- par spécification de la valeur de paramètre "ASCII437" dans le mot clé "Convert" pour les systèmes qui mettent en œuvre la page de code 437. Ces systèmes doivent le déclarer dans l'ICE;
- par spécification de la valeur de paramètre "ASCII" dans le mot clé "Convert" pour les systèmes qui mettent en œuvre la page de code 437 comme étant leur jeu de caractères local.

Autrement, seule la partie inférieure du tableau (c'est-à-dire les octets 20_{HEX} à 7E_{HEX}) a l'assurance d'être fidèlement convertie.

NOTE – Il convient de lire les tableaux suivants en choisissant d'abord une colonne, puis une ligne; par exemple, le caractère 'A' est placé dans la colonne 4, ligne 1.

8.1.2 Jeu de caractères du format de transfert ASCII normalisé APPLI/COM

Les caractères acceptés par le format de transfert ASCII normalisé APPLI/COM pour les services télétexte, télex et télécopie sont indiqués dans les Tableaux 54, 55 et 56.

8.2 Format de transfert APPLI/COM: T.61

Le format de transfert APPLI/COM T.61 correspond exactement à la Recommandation UIT-T T.61 pour le service télétexte. Les documents que l'application transfère dans ce format sont adaptés tels quels au service télétexte. Le format et les codes doivent être vérifiés mais aucune conversion ne doit être effectuée. Le format convient pour les applications qui génèrent déjà ce format ou sont destinées à transmettre des textes de présentation complexes via le service télétexte.

8.3 Format de transfert APPLI/COM: TIFF

Le format TIFF est un format de transfert orienté graphique. Le terme TIFF signifie "format de fichier image par étiquettes". Le format des fichiers TIFF comprend des attributs qui décrivent l'image tels que la résolution et les dimensions, et sont incorporés à l'aide d'étiquettes dans l'en-tête de fichier TIFF. Etant donné que l'accès aux informations se fait au moyen d'étiquettes, le programme qui engendre les fichiers TIFF (éditeur TIFF) n'est pas lié à une structure de fichier constante, car il suffit qu'un programme de lecture TIFF (lecteur TIFF) connaisse l'algorithme nécessaire pour localiser les étiquettes.

Le format de transfert TIFF présente les caractéristiques suivantes:

- il contient les informations en pixels;
- il accepte des formats de compression de données;
- il est indépendant de l'équipement du système générateur (il fonctionne quel que soit l'ordre des octets);
- il est souple en raison de sa structure par étiquettes.

Etant donné que le nombre de combinaisons possibles d'étiquettes est très élevé et que les étiquettes définies ne sont pas toutes nécessaires pour décrire une image, plusieurs classes de format TIFF ont été formées au cours de la mise au point du format TIFF. C'est pourquoi la présente Recommandation donne une définition du profil pris comme base pour un fichier TIFF compatible avec l'interface.

En tant qu'éditeur TIFF, une application de communication peut produire 4 classes de fichiers:

- le format TIFF normalisé, appelé ici classe 1; il s'agit de la classe par défaut pour un éditeur TIFF d'application de communication;
- le format TIFF avec valeur de compression 2, appelé ici classe 2. L'acceptation de ce format est une caractéristique facultative de l'application de communication;
- le format TIFF avec compression de groupe 3 de l'UIT-T, appelé ici classe 3. Ce format est accepté par toutes les applications de communication qui assurent le service de télécopie de groupe 3;
- le format TIFF avec compression de groupe 4 de l'UIT-T, appelé ici classe 4. Ce format est accepté par les applications de communication qui assurent le service de télécopie de groupe 4.

⁶⁾ Le jeu de caractères du format de transfert ASCII étendu APPLI/COM est un sous-ensemble du jeu de caractères IBM-PC; aussi est-il très adapté aux instances utilisant ces systèmes.

TABLEAU 51/T.611

Jeu de caractères du format de transfert ASCII étendu APPLI/COM pour les services de télécopie de groupe 3 et 4 (correspond au jeu de caractères ASCII 437)

Hex	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	'	p	Ç	É	á	⏏	⌞	⌚	α	≡
01			!	1	A	Q	a	q	ü	æ	í	⏏	⌞	⌚	β	±
02			"	2	B	R	b	r	é	Æ	ó	⏏	⌞	⌚	Γ	≥
03			#	3	C	S	c	s	â	ô	ú		⌞	⌚	π	≤
04		¶	\$	4	D	T	d	t	ä	ö	ñ	⌞	⌚	⌚	Σ	∫
05		§	%	5	E	U	e	u	à	ò	Ñ	⌞	⌞	⌞	σ	J
06			&	6	F	V	f	v	å	û	ª	⌞	⌞	⌞	μ	÷
07			'	7	G	W	g	w	ç	ù	º	⌞	⌞	⌞	τ	≈
08			(8	H	X	h	x	ê	ÿ	¿	⌞	⌞	⌞	Φ	°
09)	9	I	Y	i	y	ë	Ö	⌞	⌞	⌞	⌞	Θ	•
0A			*	:	J	Z	j	z	è	Ü	⌞	⌞	⌞	⌞	Ω	·
0B			+	;	K	[k	{	ï	ø	½	⌞	⌞	■	δ	√
0C			,	<	L	\	l		î	£	¼	⌞	⌞	■	∞	ⁿ
0D			-	=	M]	m	}	ì	¥	¡	⌞	⌞	■	∅	²
0E			.	>	N	^	n	~	Ä	Pₜ	«	⌞	⌞	■	€	■
0F			/	?	O	_	o		Å	f	»	⌞	⌞	■	∩	

T0823430-95/d10

Sens de conversion	Opération effectuée
Sortant	Tous les caractères indiqués dans le tableau sont acceptés dans le jeu de caractères du service de télécopie
Entrant	Le format de transfert n'est pas créé dans ce sens [sauf dans les configurations où l'application de communication a une capacité OCR (Reconnaissance optique des caractères)]
<p>NOTE – Les caractères «BARRE OBLIQUE INVERSE» (5C_{HEX}) et «ACCOLADES» (7B_{HEX} et 7D_{HEX}) qui sont souvent utilisés dans les systèmes DOS et OS/2 n'appartiennent pas au jeu de caractères du télétext. Ils ne doivent donc pas être utilisés dans les documents envoyés par l'intermédiaire de ce service.</p>	

TABLEAU 52/T.611

Jeu de caractères du format transfert ASCII étendu APPLI/COM pour le service télétext

Hex	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	'	p	Ç	É	á					
01			!	1	A	Q	a	q	ü	æ	í				β	±
02			"	2	B	R	b	r	é	Æ	ó					
03			#	3	C	S	c	s	â	ô	ú					
04		¶	\$	4	D	T	d	t	ä	ö	ñ					
05		§	%	5	E	U	e	u	à	ò	Ñ					
06			&	6	F	V	f	v	å	û	ª				μ	÷
07			'	7	G	W	g	w	ç	ù	º					
08			(8	H	X	h	x	ê	ÿ	¿					°
09)	9	I	Y	i	y	ë	Ö						•
0A			*	:	J	Z	j	z	è	Ü					Ω	·
0B			+	;	K	[k		ï	ø	½					
0C			,	<	L		l		î	£	¼					
0D			-	=	M]	m		ì	¥	¡					²
0E			.	>	N	^	n	~	Ä		«					
0F			/	?	O	_	o		Å		»					

T0823440-95/d11

Sens de conversion	Opération effectuée
Sortant	Tous les caractères indiqués dans le tableau sont acceptés dans le jeu de caractères du service télétext.
Entrant	Dans le sens de réception, il est possible de recevoir des caractères qui ne sont pas contenus dans le tableau. Ces caractères ne sont pas contenus dans le tableau. Ces caractères doivent être remplacés par des caractères qui ressemblent à l'original ou par le caractère «?» (3F _{HEX}). Le caractère FA _{HEX} n'est jamais créé dans le sens réception. Le caractère F9 _{HEX} est créé à la place de celui-ci.

TABLEAU 53/T.611

Jeu de caractères du format de transfert ASCII étendu APPLI/COM pour le service télex

Hex	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0		P		p								
01				1	A	Q	a	q								
02				2	B	R	b	r								
03				3	C	S	c	s								
04				4	D	T	d	t								
05				5	E	U	e	u								
06				6	F	V	f	v								
07			'	7	G	W	g	w								
08			(8	H	X	h	x								
09)	9	I	Y	i	y								
0A			:		J	Z	j	z								
0B			+		K		k									
0C			,		L		l									
0D			-	=	M		m									
0E			.		N		n									
0F			/	?	O		o									

T0823450-95/d12

Sens de conversion	Opération effectuée
Sortant	Tous les caractères indiqués dans le tableau sont acceptés dans le jeu de caractères du service télex. Les majuscules sont interprétées comme des minuscules.
Entrant	Dans le sens de réception, les lettres sont toujours codées en minuscules (Codes 61 _{HEX} à 7A _{HEX}). Les majuscules ne sont pas générées.

TABLEAU 54/T.611

**Jeu de caractères du format de transfert ASCII normalisé APPLI/COM
pour les services de télécopie de groupe 3 et 4**

Hex	00	10	20	30	40	50	60	70
00			SP	0	@	P	'	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			\$	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(8	H	X	h	x
09)	9	I	Y	i	y
0A			*	:	J	Z	j	z
0B			+	;	K	[k	{
0C			,	<	L	\	l	
0D			-	=	M]	m	}
0E			.	>	N	^	n	~
0F			/	?	O	_	o	

T0823460-95/d13

Sens de conversion	Opération effectuée
Sortant	Tous les caractères indiqués dans le tableau sont acceptés dans le jeu de caractères du service de télécopie de groupe 3 et 4.
Entrant	Le format de transfert n'est pas créé dans ce sens.
<p>NOTE – Les caractères «BARRE OBLIQUE INVERSE» (5C_{HEX}) et «ACCOLADES» (7B_{HEX} et 7D_{HEX}) qui sont souvent utilisés dans les systèmes DOS et OS/2 n'appartiennent pas au jeu de caractères du télétext. Ils ne doivent donc pas être utilisés dans les documents envoyés par l'intermédiaire de ce service.</p>	

TABLEAU 55/T.611

**Jeu de caractères du format de transfert ASCII normalisé APPLI/COM
pour le service télétext**

Hex	00	10	20	30	40	50	60	70
00			SP	0	@	P	'	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			\$	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(8	H	X	h	x
09)	9	I	Y	i	y
0A			*	:	J	Z	j	z
0B			+	;	K	[k	
0C			,	<	L		l	
0D			-	=	M]	m	
0E			.	>	N	^	n	~
0F			/	?	O	_	o	

T0823470-95/d14

Sens de conversion	Opération effectuée
Sortant	Tous les caractères indiqués dans le tableau sont acceptés dans le jeu de caractères du service télétext.
Entrant	Dans le sens de réception, il est possible de recevoir des caractères qui ne sont pas contenus dans le tableau ci-dessus. De tels caractères peuvent être remplacés par le caractère «?» (3F _{HEX}).

TABLEAU 56/T.611

**Jeu de caractères du format de transfert ASCII normalisé APPLI/COM
pour le service télex**

Hex	00	10	20	30	40	50	60	70
00			SP	0		P		p
01				1	A	Q	a	q
02				2	B	R	b	r
03				3	C	S	c	s
04				4	D	T	d	t
05				5	E	U	e	u
06				6	F	V	f	v
07			'	7	G	W	g	w
08			(8	H	X	h	x
09)	9	I	Y	i	y
0A			:	J	Z	j	z	
0B			+		K		k	
0C			,		L		l	
0D			-	=	M		m	
0E			.		N		n	
0F			/	?	O		o	

T0823480-95/d15

Sens de conversion	Opération effectuée
Sortant	Tous les caractères indiqués dans le tableau sont acceptés dans le jeu de caractères du service télex. Les majuscules sont interprétées comme des minuscules
Entrant	Dans le sens de réception, les lettres sont toujours codées en minuscules (61 _{HEX} à 7A _{HEX}). Les majuscules ne sont pas générées

8.3.1 Profil APPLI/COM TIFF

Les étiquettes reconnues par une application de communication et leur mode de traitement sont résumés dans le Tableau 57.

TABLEAU 57/T.611

Tableau de profil TIFF APPLI/COM

Etiquette		Unité de lecture		Unité d'écriture			
HEX	Nom	Valeur acceptée (dec)	Valeur par défaut (dec)	Valeur classe 1 (dec)	Valeur classe 2 (dec)	Valeur classe 3 (dec)	Valeur classe 4 (dec)
FE	NewSubfileType (Note 1)	"0", "2"	"0"	"0"	"0"	"0"	"0"
FF	SubfileType (Note 1)	"0", "2"	"0"	"0"	"0"	"0"	"0"
100	ImageWidth	'Largeur' (Note 2)	Rejeter	'Largeur'	'Largeur'	'Largeur'	'Largeur'
101	ImageLength	'Longueur' (Note 3)	Rejeter	'Longueur'	'Longueur'	'Longueur'	'Longueur'
102	BitsPerSample	"1"	"1"	"1"	"1"	"1"	"1"
103	Compression	"1", "2", "3", "4", "32773" (Note 4)	"1"	"1"	"2"	"3"	"4"
106	Photometric-Interpretation	"0", "1"	"0"	"0"	"0"	"0"	"0"
107	Thresholding	Ignorer		Aucune			
108	CellWidth	Ignorer		Aucune			
109	CellLength	Ignorer		Aucune			
10A	FillOrder	"1", "2" (Note 5)	"1"	"1"	"1"	"2"	"2"
10D	DocumentName	Ignorer		Aucune			
10E	ImageDescription	Ignorer		Aucune			
10F	Make	Ignorer		Aucune			
110	Model	Ignorer		Aucune			
111	StripOffsets	'Décalage'	Rejeter	'Décalage'	'Décalage'	'Décalage'	'Décalage'
112	Orientation	"1"	"1"	"1"	"1"	"1"	"1"
115	SamplesPerPixel	"1"	"1"	"1"	"1"	"1"	"1"
116	RowsPerStrip	'Lignes'	Rejeter	'Lignes'	'Lignes'	'Lignes'	'Lignes'
117	StripByteCounts	'Décompte'	Rejeter	'Décompte'	'Décompte'	'Décompte'	'Décompte'
118	MinSampleValue	"0"	"0"	"0"	"0"	"0"	"0"
119	MaxSampleValue	"1"	"1"	"1"	"1"	"1"	"1"
11A	XResolution	"300" (Note 6)	"300"	"300"	"300"	'Rés-X'	'Rés-X'
11B	YResolution	"300" (Note 6)	"300"	"300"	"300"	'Rés-Y'	'Rés-Y'

TABLEAU 57/T.611 (fin)

Tableau de profil TIFF APPLI/COM

Étiquette		Unité de lecture		Unité d'écriture			
HEX	Nom	Valeur acceptée (dec)	Valeur par défaut (dec)	Valeur classe 1 (dec)	Valeur classe 2 (dec)	Valeur classe 3 (dec)	Valeur classe 4 (dec)
11C	PlanarConfiguration	"1"	"1"	"1"	"1"	"1"	"1"
11D	PageName	Ignorer		Aucune			
11E	XPosition	Ignorer		Aucune			
11F	YPosition	Ignorer		Aucune			
120	FreeOffsets	Ignorer		Aucune			
121	FreeByteCount	Ignorer		Aucune			
122	GrayResponseUnit	Ignorer		Aucune			
123	GrayResponseCurve	Ignorer		Aucune			
124	Group3Options	"0", "4" (Note 7)	"0"	"0"	"0"	"4"	"4"
125	Group4Options	"0", "4" (Note 7)	"0"	"0"	"0"	"4"	"0"
128	ResolutionUnit	"2"	"2"	"2"	"2"	"2"	"2"
129	PageNumber	'Page'	Rejeter	'Page'	'Page'	'Page'	'Page'
12C	ColorResponseUnit	Ignorer		Aucune			
12D	ColorResponseCurve	Ignorer		Aucune			

NOTES

1 Les étiquettes NewSubfileType et SubfileType doivent l'une et l'autre être acceptées par les unités de lecture. L'unité d'écriture doit créer l'étiquette NewSubfileType dans le seul cas où elle a remplacé l'étiquette SubfileType.

2 Les règles suivantes s'appliquent: ImageWidth/XResolution (LargeurImage/RésolutionX) \leq 215 mm. Si le quotient est dépassé, le document peut être rejeté. Si la RésolutionX est de 204 dpi (norme de télécopie, G3) la valeur 1728 n'est pas dépassée pour la largeur image.

3 ImageLength est traitée si le quotient imageLength/YResolution (LongueurImage/RésolutionY) \leq 297 mm. Cette condition correspond à la longueur d'une feuille de papier A4 DIN. La garantie s'applique jusqu'à une résolution de 300 dpi. Le traitement de valeurs plus élevées de longueur d'image est une caractéristique facultative de l'application de communication.

4 Acceptation des valeurs de compression:

- 1: la valeur 1 (sans compression) n'est garantie que si la valeur de résolution (résolutionX, résolutionY) est de 300 dpi ou correspond exactement à celle du service de télécopie choisi;
- 2: la valeur 2 est une caractéristique facultative de l'application de communication;
- 3 ou 4: la compression de groupe 3/4 de l'UIT-T n'est garantie que si la valeur de résolution (résolutionX, résolutionY) correspond exactement aux résolutions des services de télécopie choisis;
- 32773: la valeur 32773 (compression d'éléments binaires en paquets) est une caractéristique facultative de l'application de communication.

5 La valeur FillOrder (ordre de remplissage) 2 (= "ordre inverse des éléments binaires") n'est autorisée que si la valeur de compression est égale à 3 ou 4.

6 Pour les valeurs de compression 3 et 4, la valeur de résolution effective (résolutionX, résolutionY) en dpi qui correspond exactement à celle du service de télécopie choisi doit être spécifiée. La valeur par défaut de l'unité de lecture n'est pas applicable étant donné que les résolutions X et Y doivent être appliquées. L'application locale est chargée de la conversion locale des résolutions reçues.

7 Si la valeur de compression est 3, la valeur de Group3Options est 4; sinon cette valeur est nulle. Toute autre combinaison est rejetée.

Le Tableau 58 résume les règles applicables au tableau de profil TIFF:

TABLEAU 58/T.611

Règles applicables au tableau de profil TIFF

Colonne	Règle
Étiquette	Indique le nom de l'étiquette et sa valeur en notation hexadécimale. Si une unité de lecture TIFF rencontre une étiquette ne figurant pas dans la liste, elle peut rejeter le fichier TIFF
Unité de lecture – Valeur acceptée	Résume les valeurs acceptées. Seule est garantie l'acceptation par l'application de communication des valeurs énumérées. Les autres valeurs peuvent être rejetées par une application de communication et doivent donc être évitées par une application locale génératrice. Si une valeur indique "ignorer", l'unité de lecture ne tient pas compte de l'étiquette
Unité de lecture – Valeur par défaut	Indique la valeur par défaut appliquée si l'étiquette n'est pas spécifiée. Lorsqu'une valeur indique "rejeter", le fichier TIFF doit être rejeté si l'étiquette est absente. Si une valeur indique "ignorer", aucune valeur par défaut n'est applicable car l'unité de lecture ne tient pas compte de l'étiquette
Unité d'écriture	Les colonnes unité d'écriture indiquent les valeurs créées par la classe appropriée d'unité d'écriture TIFF d'application de communication. Si une valeur indique "aucune", aucune étiquette ni valeur n'est créée dans cette classe

8.4 Contraintes de service s'appliquant aux formats de transfert

Etant donné la nature des services de télécommunication choisis, plusieurs contraintes s'appliquent aux formats de transfert. Ces contraintes sont résumées ci-dessous dans le Tableau 59.

TABLEAU 59/T.611

Contraintes de format de transfert

Service de télécommunication	Format de transfert	Contraintes
Télex	ASCII, T.50	Pas de spécifications de format mais le balisage du texte est autorisé. Aucune autre spécification ne doit être prise en compte. Certaines contraintes s'appliquent également au jeu de caractères
Télétext	ASCII, T.50, T.61	Seules les spécifications de format propres au télétext sont autorisées. Toutes les autres spécifications ne sont pas prises en compte ou le fichier est rejeté. Des contraintes s'appliquent également au jeu de caractères
	VOID	Autorisé seulement conjointement avec le choix d'un type de format TFT (transfert de fichier télématique)
Télécopie (G3/G4)	ASCII, T.50 ^{a)}	Tous les formats et attributs sont autorisés. Si une application de communication n'accepte pas un attribut ou un format spécifique, elle est autorisée à ne pas en tenir compte
	TIFF	Autorisé seulement si aucun des types de format TFT (transfert de fichier télématique) n'a été choisi
	VOID	Autorisé seulement conjointement avec le choix d'un type de format TFT (transfert de fichier télématique)
^{a)} Seulement applicable dans le sens sortant.		

9 ICE (environnement de configuration d'interface)

Avant qu'une application locale puisse commencer à dialoguer avec une application de communication, l'application locale doit savoir quelle est l'application de communication dont elle a besoin (d'après des critères tels que le type de services de télécommunication pris en charge) et comment "parler" à cette application de communication. Etant donné que l'interface décrite dans la présente Recommandation est destinée à plusieurs plates-formes, le mécanisme recommandé pour obtenir des informations des applications de communication avant de dialoguer avec elles repose sur la lecture de l'information de configuration. Cette information de configuration est appelée l'environnement de configuration d'interface (ICE).

Un fournisseur d'application de communication doit fournir cette information de configuration, afin que les applications locales puissent obtenir des informations sur une application de communication spécifique ou sur la manière d'y accéder.

La partie de l'environnement ICE qui donne des informations sur une application de communication spécifique est appelée le descripteur de CA. Selon que le système est un système autonome ou qu'il utilise un réseau de zone locale (RZL), les informations concernant les applications de communication accessibles se trouvent en différents endroits. Etant donné que plusieurs applications de communication sont accessibles par une application locale, il est nécessaire de centraliser les emplacements des descripteurs de CA individuels dans un fichier d'"environnement ICE maître". L'emplacement de l'environnement du fichier d'environnement ICE maître proprement dit est défini pour chaque plate-forme (voir l'Annexe B).

Par conséquent, sur un système donné, une application locale doit d'abord lire le fichier de l'environnement ICE maître pour découvrir les applications de communication auxquelles elle peut accéder et l'emplacement des descripteurs de CA de ces applications de communication. L'application locale lit ensuite le descripteur de CA de l'application de communication qu'elle souhaite utiliser. Enfin, l'application locale ouvre une session avec cette application de communication en utilisant l'information trouvée dans le descripteur de CA de cette application de communication.

Pour illustrer ces concepts, trois configurations types peuvent être envisagées:

- un système autonome prenant en charge plusieurs applications locales et plusieurs applications de communication;
- un système utilisant un réseau de zone locale (RZL), les applications locales et les applications de communication se situant sur des machines différentes, avec serveur de fichiers;
- un système utilisant un réseau de zone locale (RZL), les applications locales et les applications de communication se situant sur des machines différentes, sans serveur de fichiers.

Dans le cas a), l'environnement ICE maître est situé dans un chemin de fichier spécifique (dont l'emplacement précis dépend de la plate-forme, voir l'Annexe B), et l'ICE maître donne la liste de toutes les applications de communication sur ce système avec l'emplacement de leur descripteur de CA respectif (voir la Figure 12).

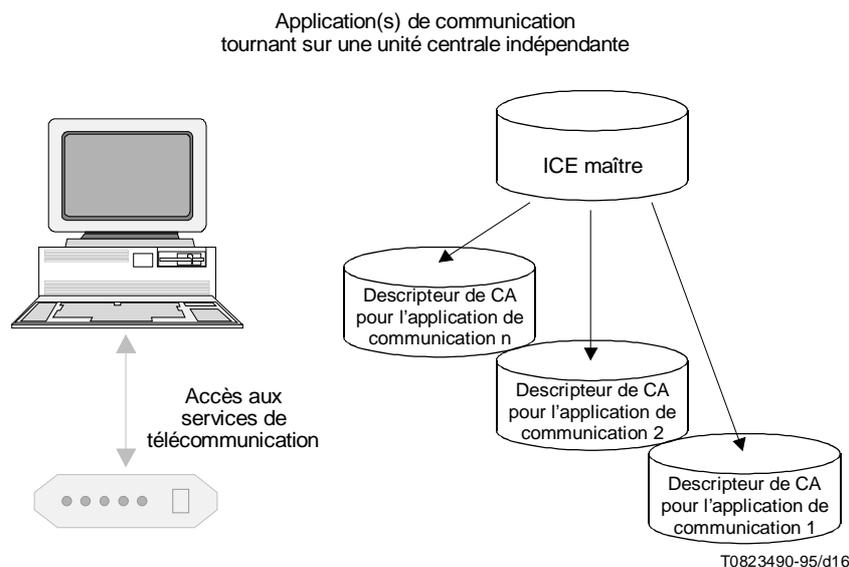


FIGURE 12/T.611

Système autonome acceptant plusieurs applications de communication

Dans le cas b), l'ICE maître se trouve dans le serveur de fichiers, dans un chemin de fichier auquel tous les postes de saisie peuvent accéder. L'ICE maître donne la liste de toutes les applications de communication du réseau de zone locale (RZL) et indique les chemins absolus de leur descripteur de CA respectif (qui peuvent se trouver sur d'autres machines que le serveur de fichiers proprement dit) (voir la Figure 13).

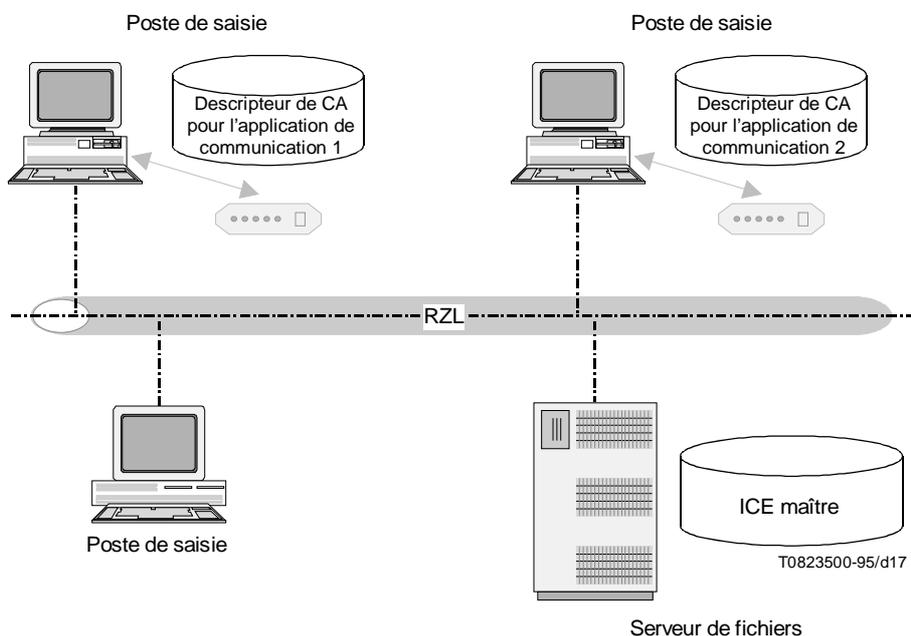


FIGURE 13/T.611

Système utilisant un réseau de zone locale (RZL) avec serveur de fichiers

Dans le cas c), aucun serveur de fichiers ne pouvant centraliser le fichier de l'ICE maître pour l'ensemble du système, un fichier d'ICE maître se trouve sur chaque machine acceptant une ou plusieurs applications de communication, en un emplacement fixe comme dans le cas a). L'emplacement des descripteurs de CA est enregistré dans l'ICE maître avec les chemins relatifs à cette machine (voir la Figure 14).

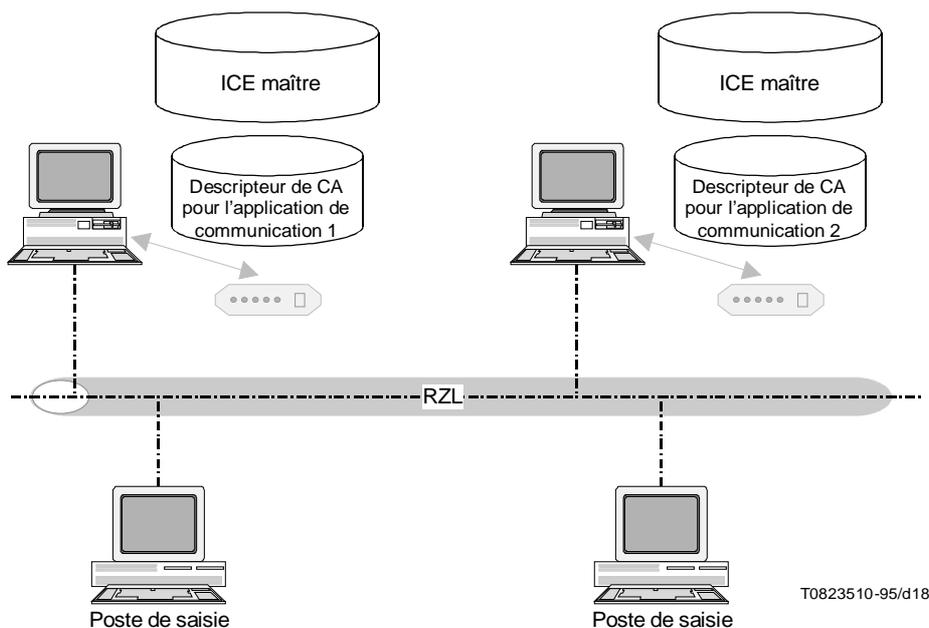


FIGURE 14/T.611

Système de réseau de zone locale (RZL) homologue à homologue

9.1 Présentation de l'ICE

L'ICE se présente sous forme de caractères codés en mode texte lisible, tel que décrit pour les descriptions TDD du 6.4.1. A l'exception des systèmes où le codage naturel des caractères est un codage EBCDIC (identificateur Code-ID mis à "E"), le codage IRA est utilisé (identificateur Code-ID mis à "I"). L'ICE (ICE maître et descripteurs de CA) doit se présenter sous la forme de lignes utilisant le répertoire des caractères IRA et être formaté comme décrit dans 6.4.1 concernant les descriptions TDD.

9.2 Obtention de l'accès à l'ICE

L'accès à l'ICE maître est obtenu grâce à une "méthode d'accès de fichier" commune à la plupart des systèmes d'exploitation. En retour, toutes les méthodes disponibles pour obtenir l'information de descripteur de CA (CA-Descriptor) pour une application de communication donnée doivent être définies dans l'ICE maître. Les méthodes d'accès peuvent être "basées sur l'accès de fichier" ou peuvent profiter de méthodes plus dynamiques disponibles dans un environnement d'exploitation particulier (à savoir bibliothèque de liens dynamiques DLL pour Windows et fichiers exécutables ou pilotes de périphériques pour DOS).

9.3 ICE maître

L'ICE maître donne la liste de toutes les applications de communication auxquelles une application locale peut accéder. Il contient des informations sur l'identificateur de CA, les services de communication assurés, le type de mécanisme d'échange utilisé et l'emplacement du descripteur de CA de l'application de communication où des informations supplémentaires peuvent être obtenues.

9.3.1 Description formelle

Le présent paragraphe décrit la syntaxe de l'ICE maître. Le premier élément de syntaxe dans l'ICE maître ou dans le descripteur de CA est l'en-tête <Master ICE Header>. Aucun autre élément, pas même un opérateur de formatage par caractère SPACE ou TABULATION, ne doit précéder l'en-tête. La syntaxe détaillée en forme de type BNF, est montrée ci-dessous (voir A.1 pour une description de la forme de type BNF):

```
<Master ICE> ::=          <Master ICE Header> <CA Entries>

<Master ICE Header> ::=   "I*APPLI/COM*" <Version> "*ITU-T*MASTER_ICE"
-- pour la définition de l'en-tête <ICE Header> voir aussi 6.3.2.3

<Version> ::=            "1994"
-- <Version> désigne l'année d'approbation de la présente Recommandation

<CA Entries> ::=         "#" <CA Entry> { "#" <CA Entry> }

<CA Entry> ::=           <CA Identifieur> <CA Product Info> <CA EM>
                        <CA Service> { <CA Service> }
                        <CA ICE Location> { <CA ICE Location> }

<CA Identifieur> ::=     "CA-ID" ":" NUMERIC-STRING
-- identifie l'application de communication

<CA Product Info> ::=   "APPLI/COM" ":" STRING(SIZE(1..255))
-- Information concernant le produit, fournie par le fabricant de
-- l'application de communication

<CA EM> ::=              "EM" ":" "FILE" | "PRIMITIVE"
-- mécanisme d'échange assuré par l'application de communication
-- voir 7.3

<CA Service> ::=        "SERVICE" ":" <Service-id-parameter>
-- services de télécommunication assurés par l'application de communication
-- voir 6.4.5.1 et 10.4

<CA ICE Location> ::=   "ACCESS" ":" PATH
-- indique l'emplacement du fichier de descripteurs de CA ou du module de pilotage
```

9.3.2 Configuration

L'ICE maître peut être configuré manuellement au moyen d'un logiciel "éditeur" lors de l'installation d'une application de communication particulière ou dynamiquement au moyen d'un logiciel de "pilotage" approprié, fourni par le fabricant de l'application de communication⁷⁾.

L'ICE doit être configuré de manière à tenir compte de toutes les applications de communication accessibles à partir d'un système⁸⁾.

9.4 Descripteur de CA

Le descripteur de CA contient les informations relatives à une application de communication donnée. Il est constitué de sections regroupant des éléments liés logiquement entre eux.

9.4.1 Description formelle

Le présent paragraphe décrit la syntaxe du descripteur de CA. Le premier élément de syntaxe du descripteur de CA est l'en-tête <CA-Descriptor Header>. Aucun autre élément, pas même un opérateur de formatage par caractère SPACE ou TABULATION, ne doit précéder l'en-tête. La syntaxe détaillée, en formalisme de type BNF, est montrée ci-dessous:

```
<CA-Descriptor> ::=          <CA-Descriptor Header> <CA-Descriptor-component>
<CA-Descriptor Header> ::=  "I*APPLI/COM*" <Version> "ITU-T*ICE"
                             -- pour la définition de l'en-tête <CA-Descriptor Header> voir aussi 6.3.2.3
<Version> ::=                "1994"
                             -- <Version> désigne l'année d'approbation de la présente Recommandation
<CA-Descriptor-component> ::= "#" <Section> {<Section>}
<Section> ::=                <Section-Parameter> {<Section-Parameter>}
<Section-Parameter> ::=      <Parameter-Name> ":" <Parameter-Value>
<Parameter-Name> ::=        STRING(SIZE(1..16))
<Parameter-Value> ::=       STRING(SIZE(1..255))
```

9.4.2 Configuration

L'information de descripteur de CA est fournie par le fabricant de l'application de communication sous forme de fichier ou par des moyens dynamiques tels qu'un pilote, un fichier exécutable ou une bibliothèque de liens dynamiques (DLL). La(les) méthode(s) d'accès pour le descripteur de CA doi(ven)t être définie(s) dans l'ICE maître en tant que partie de la procédure d'installation pour chaque application de communication. Le descripteur de CA peut être configuré manuellement au moyen d'un logiciel "éditeur" lors de l'installation d'une application de communication particulière ou dynamiquement au moyen d'un logiciel de "pilotage" approprié, fourni par le fabricant de l'application de communication. Cependant, une application locale qui compte sur les informations émanant d'un descripteur de CA pour une application de communication particulière ne doit pas supposer que l'application de communication est active (chargée et en fonctionnement) à cet instant. Pour être certaine qu'une application de communication est réellement active, l'application locale doit ouvrir une session avec cette application de communication. L'application de communication n'est considérée comme active que lorsque l'ouverture de session réussit. Pour plus d'information concernant la procédure d'ouverture de session, voir 7.1 et 7.2.1.

9.5 Composantes du descripteur de CA

Les Tableaux 60, 61 et 62 indiquent les mots clés pertinents et les valeurs de paramètres du descripteur de CA correspondantes des enregistrements de CA. Le paragraphe 6.4.4 donne des renseignements complémentaires sur le codage des valeurs de paramètres.

⁷⁾ Etant donné que l'accès aux fichiers et aux modules de pilotage des systèmes d'exploitation est similaire sur les systèmes d'exploitation courants, l'ICE maître proprement dit peut être mis en œuvre statiquement comme un fichier ou dynamiquement sous la forme d'un module de pilotage de système d'exploitation.

⁸⁾ Il n'est pas nécessaire qu'une application de communication fonctionne avec le même équipement que l'application locale. L'application de communication est tenue uniquement d'assurer l'accès à l'aide d'un module de pilotage ou d'un module similaire.

9.5.1 Paramètres de la section DESCRIPTION GÉNÉRALE

Voir le Tableau 60.

TABLEAU 60/T.611

Eléments d'information du descripteur de CA indépendants du mécanisme d'échange et du système d'exploitation

Mot clé ^{a)}	Paramètre	Interprétation
ALIAS	"oui" "non"	Spécifie si l'application de communication prend en charge les "alias" (pseudonymes ou noms conviviaux)
CODEPAGE*	STRING	Spécifie les pages de code supplémentaires décrivant les jeux de caractères ASCII étendu qui sont acceptés par l'application de communication. Le paramètre de type chaîne indique le numéro de la page de code, par exemple "850"
CODING*	Code-ID	Spécifie quel type de codage des descriptions TDD est accepté par l'application de communication. Voir 6.3 pour les valeurs acceptées qui peuvent être spécifiées. Si la valeur "E" de l'identificateur Code-ID apparaît dans l'ICE, sa valeur binaire doit être choisie en accord avec la présentation de code choisie pour l'ICE lui-même, c'est-à-dire si le paramètre Code-id de l'en-tête ICE est "I" (ASCII normalisé APPLI/COM), alors la valeur binaire de "E" doit être codée à 45 _{HEX}
CONVCHK*	<Convert-id-parameter>	Déclare quels formats de transfert sont acceptés dans la fonction Submit TDD CONVERT et/ou CHECK. Pour la syntaxe du paramètre <Convert-id-parameter>, voir également 6.4.5.3
COUNTRY	STRING	Spécifie le pays pour lequel l'application de communication a été configurée. Utilisé pour enregistrer les caractéristiques propres au pays considéré, telles que l'obtention d'un accès au service de conversion ou d'un accès à la liste noire de numéros d'appel. La valeur à placer dans ce paramètre est à prendre dans la Recommandation UIT-T T.35. Elle doit se présenter sous forme de chaîne numérique, en base décimale; par exemple "154" pour les Seychelles
DRF	"oui" "non"	Paramètre booléen, qui indique si l'application de communication offre le service de "distribution des fichiers reçus"
EXTEND*	<keyword>	Permet l'apport d'extensions à la Recommandation. Ne peut être mis en œuvre que sous forme de modifications de la Recommandation. Tous les mots clés acceptés par l'application de communication doivent être énumérés
FC	"A" "B"	Indique quelle classe fonctionnelle est acceptée par l'application de communication
FONT0	NUMERIC-STRING	Nombre compris entre 1 et 9 pour spécifier le nombre de polices déclarées. Les polices sont décrites de l'identificateur "Font1" à l'identificateur "Font9" (voir ci-dessous)
FONT _x *	STRING "," NUMERIC-STRING	Nom de la police, suivi de son corps (en caractères par pouce). La valeur de "x" varie entre 1 et 9
NATIONAL*	<keyword>	Permet l'apport d'extensions nationales à la Recommandation. Ne peut être mis en œuvre qu'avec le consentement des Administrations nationales. (Tous les mots clés acceptés doivent être énumérés)
PRINT*	<Printer-id-parameter>	Déclare à quelles imprimantes l'application de communication peut accéder au titre de la fonction Submit TDD PRINT. Pour la syntaxe du paramètre <Printer-id-parameter>, voir également 6.4.4.10
PRIVATE*	<keyword>	Permet l'apport d'extensions privées à la Recommandation. (Tous les mots clés acceptés doivent être énumérés)
RECORD*	<keyword> "," NUMERIC-STRING	Donne la liste complète des noms de champ d'enregistrement CA acceptés par l'application de communication dans l'ordre où ils se trouvent dans le fichier à la suite de la fonction Trace:COPY. Le fabricant de l'application de communication devra indiquer le mot clé suivi par la longueur – séparée par des virgules – que le champ aura dans le fichier résultant. Voir également le Tableau 23
SUBMIT*	"PRINT" "CONVERT" "CHECK"	Déclare quelles fonctions sont assurées au titre de la fonction Submit TDD. Ce mot clé doit être répété autant de fois que nécessaire

^{a)} Les mots clés dont le nom est suivi d'un astérisque (*) peuvent être répétés.

9.5.2 Paramètres de la section méthode d'échange (EM)

Voir le Tableau 61.

TABLEAU 61/T.611

Éléments d'information du descripteur de CA s'appliquant au mécanisme d'échange de fichiers

Mot clé	Paramètre	Interprétation
EM	"file" "primitive"	Méthode d'échange utilisée pour l'échange de descriptions TDD entre applications locales et applications de communication. "fichier" ou "primitive" sont les valeurs acceptées (voir 7.3 pour plus de détails)
SYNC	"oui" "non"	Indique si la CA est "pilotee-sync". Voir 7.3 pour plus de détails
F_JOB_Q	PATH	Spécifie le chemin d'accès aux fichiers de demandes de descriptions TDD. Voir 7.3 pour plus de détails
F_ACK_Q	PATH	Spécifie le chemin d'accès aux fichiers de réponses de descriptions TDD. Voir 7.3 pour plus de détails
ERROR_Q	PATH	Spécifie le chemin d'accès aux fichiers de réponses de descriptions TDD se rapportant aux erreurs. Voir 7.3 pour plus de détails
ALARM	"oui" "non"	Ce paramètre booléen établit si l'application de communication prend en charge la fonction SetAlarm

9.5.3 Paramètres de la section SYSTÈME D'EXPLOITATION

Voir le Tableau 62.

9.5.4 Paramètres de la section SERVICES

Voir le Tableau 63.

D'autres éléments propres aux services sont décrits dans les articles concernés de la Partie II de la présente Recommandation.

10 Classes et profils fonctionnels

10.1 Classe fonctionnelle A

La classe fonctionnelle A (FCA) spécifie un ensemble de formats de transfert ainsi que les interactions entre application locale et application de communication ayant pour but d'émettre ou de recevoir des documents.

La classe FCA exige que les fonctions suivantes soient prises en charge:

- les fonctions <SendTDD> et <SendAckTDD> qui permettent à une application locale d'envoyer un document par l'intermédiaire d'une application de communication;
- la fonction <ReceiveTDD> qui permet à une application locale de récupérer des documents reçus par une application de communication.

En outre, l'application locale peut choisir diverses options offertes par l'application de communication telles que l'utilisation de mots clés supplémentaires ("ADDKEYS") ou la fonction de soumission ("SUBMIT").

10.2 Classe fonctionnelle B

En plus des caractéristiques de la FCA, la classe fonctionnelle B (FCB) assure entièrement la fonction Trace (suivi).

La FCB est un surensemble de la FCA; elle offre une intégration plus poussée des fonctions de communication aux applications d'utilisateur.

TABLEAU 62/T.611

Éléments du descripteur de CA s'appliquant au système d'exploitation

Mot clé ^{a)}	Paramètre	Interprétation
ENVIRON*	"MSDOS" "WINDOWS" "UNIX" "OS2" "MacOS"	Ce mot clé spécifie l'environnement d'exploitation de l'application de communication. Si l'application prend en charge plusieurs environnements, l'ICE doit contenir autant de mots clés ENVIRON qu'il y a de systèmes d'exploitation pris en charge
DRIVER	PATH	Nom du module de pilotage qui doit être ouvert pour engager des dialogues avec l'application de communication. Voir l'Annexe E pour plus de détails
INT	HEX,HEX	Indique le numéro d'interruption. Se compose de deux nombres hexadécimaux, le premier spécifiant le numéro du multiplex, le deuxième le numéro de code de programme. Si l'interruption n'est pas multiplexée, il n'est pas nécessaire de spécifier le deuxième nombre hexadécimal
LIB	"oui" "non"	Ce paramètre booléen établit si l'application de communication est une bibliothèque statique (l'application locale doit lui être liée)
LIB-NAME*	PATH	Chemin(s) d'accès à la (aux) bibliothèque(s) (utilisé conjointement avec le mot clé LIB)
DLL	"oui" "non"	Bibliothèque de liens dynamiques. Voir l'Annexe E. Le mot clé "DLL-NAME" ne doit être accepté que si le mécanisme d'échange DLL est accepté
DLL-NAME*	PATH	Chemin(s) d'accès au(x) fichier(s) DLL (utilisé conjointement avec le mot clé DLL)
DDE	"oui" "non"	Mécanisme d'échange de données dynamique. Dans l'environnement WINDOWS, si l'application accepte le mécanisme d'échange DDE, elle doit spécifier "oui". Voir l'Annexe E. Les trois mots clé suivants doivent être inclus dans l'ICE, si le mécanisme DDE est utilisé
WIN-APP	STRING	Nom d'application (format MsDos) XXXXXXXX.XXX
SUBJECT*	STRING	Tous les "sujets" de l'application de communication doivent être mentionnés (le cas échéant), sinon laisser vide (à utiliser avec le mot clé DDE)
ITEM*	STRING	Tous les "éléments" de l'application de communication doivent être mentionnés (le cas échéant), sinon laisser vide (à utiliser avec le mot clé DDE)
^{a)} Les mots clés dont le nom est suivi d'un astérisque (*) peuvent être répétés.		

TABLEAU 63/T.611

Éléments du descripteur de CA s'appliquant à la section services

Mot clé ^{a)}	Paramètre	Interprétation
ADDCONV*	STRING (SIZE (1..8))	Donne la liste des formats de transfert supplémentaires fournis par l'application de communication. Les noms cités ici doivent être utilisés pour sélectionner les formats de transfert aux endroits appropriés
ADDKEYS*	<keyword>	Enumère tous les mots clés supplémentaires acceptés par la CA. Seuls les mots clés de la classe "+" des tableaux de descriptions TDD de l'article 6 peuvent être spécifiés ici
^{a)} Les mots clés dont le nom est suivi d'un astérisque (*) peuvent être répétés.		

10.3 Fonctions supplémentaires

Les fonctions Submit et Extend sont des fonctions supplémentaires qui doivent être déclarées par l'application de communication dans l'ICE (voir également 9.5).

10.4 Profils de services

La prise en charge de toutes les caractéristiques d'un service de télécommunication donné peut s'avérer impossible en raison d'un trop grand nombre de fonctions à mettre en œuvre. D'un autre côté, une offre d'options trop nombreuses pour la prise en charge d'un service donné conduit à réaliser des instances conformes, mais incompatibles.

Pour réduire les incompatibilités entre les diverses applications commercialisées, la présente Recommandation prend en charge le concept de "profils de service". Un profil de service regroupe un ensemble défini de caractéristiques de service qui doivent être assurées par les instances revendiquant la conformité à la présente Recommandation.

Cet ensemble de caractéristiques à assurer, pour un profil de service donné, est défini pour chaque service.

Une instance peut ainsi revendiquer la conformité à un profil de service et prendre en charge des caractéristiques supplémentaires. De cette manière, il est garanti que deux instances (par exemple une application locale et une application de communication), assurant le même profil de service, pourront offrir toutes les caractéristiques mentionnées dans le profil de service et offrir l'option de caractéristiques supplémentaires non obligatoires pour ce profil de service.

Les applications de communication doivent déclarer dans l'ICE maître les profils de service pris en charge (voir 9.5).

PARTIE II – ENTITÉS DÉPENDANTES DU SERVICE

11 Service: Télécopie du groupe 3

Parallèlement aux mots clés propres au service de télécopie pour les fonctions <SendTDD>, <SendAckTDD> et <ReceiveTDD>, il existe une autre fonction définie comme suit: Poll – interrogation d'un télécopieur distant concernant un document à récupérer. Cette fonction doit être mise en œuvre au moyen de la fonction <ExtendTDD>. L'extension utilisée à cette fin est appelée <PollExtension>.

De même que la fonction <PollExtension>, les fonctions <SendTDD> et <SendAckTDD> peuvent également faire l'objet d'extensions pour exécuter une opération d'interrogation Poll pendant l'envoi d'un document.

Les extensions sont décrites dans le présent article.

11.1 Éléments syntaxiques propres au service

```
<ServiceDependentKeywordsSend> :=  
    ((<Recipient> [<PollSendExtension>]) | <RecipientSpec>)  
    ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
    [<SubAddress>] [<G3Speed>] [<GenCil>] [<HighRes>] [<UseEcm>]
```

```
<ServiceDependentKeywordsSendack> :=  
    <Recipient> [<PollSendExtension>]  
    ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
    [<SubAddress>] [<G3Speed>] [<GenCil>] [<HighRes>] [<UseEcm>]
```

```
<ServiceDependentKeywordsReceive> :=  
    [<Originator>]  
    ((<Document> <Convert> <Type>) | <DocumentSpec>)  
    [SubAddress] [<CvFax3>] [<G3Speed>]
```

```
<ExtendSubFunctionKeywords> :=  
    <PollExtension>
```

```
<PollSendExtension> :=    <DoPoll> <PollPassword> [<PollSelector>]
```

```
<PollExtension> :=    <PollSubFunction> <Recipient> <PollPassword> [<PollSelector>] [<SendTime>]  
    [<ComId>] [<Minor>] [<Warning>]
```

Voir le Tableau 64.

11.2 Codage de type texte

11.2.1 Correspondance des mots clés

Voir le Tableau 65.

11.2.2 Codage des paramètres

Voir également 6.4.4 pour le codage des paramètres non liés au service utilisés.

11.2.2.1 Paramètre Service-id

Le paramètre Service-id est codé comme une chaîne STRING prenant la valeur constante "FX3".

Syntaxe:

```
<Service-id-parameter> :=    "FX3"
```

TABLEAU 64/T.611

Eléments syntaxiques additionnels pour la télécopie du groupe 3

Elément syntaxique	Objet
<Convert>	Spécifie le format de transfert à utiliser
<CvFax3>	Spécifie le format de transfert pour les fichiers reçus
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception
<DoPoll>	Spécifie si le destinataire doit être interrogé dans une fonction <SendTDD> ou <SendAckTDD>
<G3Speed>	Spécifie la rapidité de modulation souhaitée (en entrée) ou celle qui a été utilisée (en sortie)
<GenCil>	Spécifie l'insertion d'une ligne CIL lors de la production de la télécopie sortante
<HighRes>	Indique la résolution
<Minor>	Cet élément syntaxique renvoie un code d'erreur supplémentaire dans la description TDD Response
<Originator>	Spécifie l'adresse de communication de l'expéditeur
<PollPassword>	Spécifie le mot de passe pour le dispositif d'interrogation distant
<PollSelector>	Permet au destinataire d'effectuer une interrogation relative à un document particulier
<PollSubFunction>	Spécifie la sous-fonction de la fonction Poll <ExtendTDD>
<Recipient>, <RecipientSpec>	Spécifie la ou les adresses de communication du ou des destinataires
<SubAddress>	Spécifie la sous-adresse de l'expéditeur
<To>	La transmission doit se terminer au numéro de page spécifié
<Type>	Spécifie le format de transmission utilisé
<UseEcm>	Indique que l'on veut utiliser le mode correction d'erreur (entrée) ou que ce mode a été activé (sortie)
<Warning>	Cet élément syntaxique renvoie un code d'alerte additionnel dans la description TDD Response

11.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"STD"	Service de télécopie de base du groupe 3 (MH)
"BTM"	Transfert de fichier télématique (TFT) du service de télécopie du groupe 3: mode transparent de base
"DTM"	Transfert de fichier télématique (TFT) du service de télécopie du groupe 3: mode document transparent
"EDI"	Transfert de fichier télématique (TFT) du service de télécopie du groupe 3: Edifact
"BFT"	Service de télécopie du groupe 3: transfert de fichier binaire

NOTE – Le format de transmission STD (= code de Huffman modifié, MH) est défini dans la Recommandation UIT-T T.4, les formats TFT, BTM, DTM et EDI dans les Recommandations UIT-T T.571 et T.30 (Annexes) et le format de transmission BFT dans la Recommandation UIT-T T.434.

Syntaxe:

<Type-id-parameter> := "STD" | "BTM" | "DTM" | "BFT" | "EDI"

TABLEAU 65/T.611

Codage de type texte des éléments syntaxiques additionnels pour la télécopie du groupe 3
(Le signe ↵ est le caractère "ligne suivante")

Élément de syntaxique	Paire mot clé/paramètre
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvFax3>	"CVFAX3" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<DoPoll>	"DOPOLL" ":" <Boolean-parameter> ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<G3Speed>	"G3SPEED" ":" <G3-speed-parameter> ↵
<GenCil>	"GENCIL" ":" <Boolean-parameter> ↵
<HighRes>	"HIGHRES" ":" <Resolution-parameter> ↵
<Minor>	"MINOR" ":" <Error-parameter> ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<PollPassword>	"PASSWORD" ":" <Password-parameter> ↵
<PollSelector>	"POLLSELECT" ":" <Poll-select-parameter> ↵
<PollSubFunction>	"SUBFUNC" ":" "Poll" ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UseEcm>	"USEECM" ":" <Boolean-parameter> ↵
<Warning>	"WARNING" ":" <Error-parameter> ↵

11.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"ASCII"	ASCII étendu APPLI/COM (pour les fichiers sortants seulement)
"ASCII437"	ASCII étendu APPLI/COM (pour les fichiers sortants seulement)
"T.50"	ASCII étendu APPLI/COM (pour les fichiers sortants seulement)
"TIFF"	Format de transfert TIFF défini au 8.3
"TIFF2"	Format TIFF de classe 2 APPLI/COM TIFF spécifié dans 8.3 (pour les fichiers entrants seulement)
"TIFF3"	Format TIFF de classe 3 APPLI/COM TIFF spécifié dans 8.3 (pour les fichiers entrants seulement)
"VOID"	Pas de conversion à effectuer

NOTE – Si une application locale réclame un document de télécopie qui a été reçu, il est possible:

- d'obtenir le document dans le format TIFF de classe 2 APPLI/COM en spécifiant TIFF2;
- d'obtenir le document dans le format TIFF de classe 3 APPLI/COM en spécifiant TIFF3;

pour autant que l'application de communication CA soit capable de produire ce format TIFF particulier. Si seul le format TIFF est spécifié, le document sera remis dans la classe TIFF APPLI/COM par défaut (classe 1). Toutefois, dans le sens émission, il suffit uniquement de spécifier le format TIFF (sans extension de numéro), car l'information de compression est contenue dans le format de transfert lui-même.

L'utilisation du paramètre Convert-id dépend aussi de l'identificateur Type-id choisi (voir le Tableau 66).

TABLEAU 66/T.611

Valeurs attribuables autorisées pour le paramètre Convert-id dépendant de l'identificateur Type-id et du sens du trafic

Type-id	Convert-id pour le trafic sortant	Convert-id pour le trafic entrant
STD	ASCII, ASCIIxxx ^{a)} , T.50, TIFF	TIFF, TIFFx ^{b)}
BTM, DTM, BFT, EDI	VOID	
<p>a) xxx est un paramètre code-page, déclaré dans l'ICE; ainsi, ce paramètre est ASCII437 si le code-page 437 a été déclaré.</p> <p>b) x désigne la classe de format TIFF à lire, sa valeur est comprise entre 2 et 3.</p>		

Syntaxe:

```

<Convert-id-parameter> := <Convert-std> | <Convert-bin>
<Convert-std> := <Convert-std-in> | <Convert-std-out>
<Convert-std-in> := "TIFF" | "TIFF2" | "TIFF3"
-- documents entrants
<Convert-std-out> := <Ascii> | "T.50" | "TIFF"
-- documents sortants
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
<Convert-bin> := "VOID"

```

11.2.2.4 Paramètres File-of-addrspec et Address

Le paramètre File-of-addrspec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Addrspec, contenant le paramètre Address.

Le paramètre Address est codé comme une chaîne STRING. La chaîne STRING doit contenir le numéro de téléphone. Si ce numéro commence par un "!" il peut contenir des caractères spéciaux qui sont traités comme des opérateurs (ou modificateurs) de préférence à des chiffres à composer (voir le Tableau 67).

Le numéro de téléphone peut se terminer par un caractère "#" (signe de numéro) et peut être suivi par une sous-adresse (sous-adresse de l'équipement distant). Le codage de la sous-adresse est défini au 11.2.2.7.

On peut aussi donner un nom de substitution à la place du numéro de téléphone, mais il faut alors que ce nom de substitution soit précédé du caractère "&". On suppose que l'application de communication sait décoder le nom de substitution spécifié. On procède généralement de cette façon lorsque l'application de communication met en œuvre un annuaire.

TABLEAU 67/T.611

Modificateurs de la chaîne de numérotation pour le RTPC

Caractère composé	Type	Explication
"0" ... "9" "*" "#"	Chiffre à composer	Composé sans interprétation
!"	Modificateur	Premier caractère de la séquence de numérotation: passer au mode "numérotation brute"
;"	Modificateur	Pause dans le processus de numérotation. La durée dépend de l'application de communication. Il convient de noter que le caractère ";" doit être précédé du caractère d'échappement "\", sinon l'application de communication peut comprendre qu'il s'agit d'un caractère d'introduction de commentaire
:"	Modificateur	Même explication que pour ";" mais n'a pas besoin d'être précédé par un caractère d'échappement
,	Modificateur	Pause de numérotation de 2 secondes
"T" "t"	Modificateur	Passage à la numérotation fréquentielle pour les chiffres suivants
"P" "p"	Modificateur	Passage à la numérotation par impulsions pour les chiffres suivants
"W" "w"	Modificateur	L'application de communication se place en attente d'une tonalité d'invitation à numéroté de 3 secondes
"@"	Modificateur	L'application de communication se place en attente d'un signal de sonnerie distant suivi d'un silence de 5 secondes. En l'absence de réponse distante, l'application de communication agit conformément aux actions par défaut
NOTE – Les modificateurs de la chaîne de numérotation peuvent faire l'objet d'accord entre Administrations nationales.		

Syntaxe:

<File-of-addrspec> := PATH
 -- *Le trajet pointe sur un fichier contenant un ou plusieurs paramètres*
 -- <Addr-spec-parameter>

<Addr-spec-parameter> := <Address-parameter>
 ["," <Cover-path> "," <Cover-conv> ["," <Cover-type>] ["," <Nopgbrk>]]

<Address-parameter> := (<Phone-number> ["#" <Subaddress>]) |
 ("!" <Dial-command> ["#" <Subaddress>]) |
 ("&" <Alias>)
 -- *contient la séquence de numérotation téléphonique ou le nom de*
 -- *substitution (alias)*

<Phone-number> := NUMERIC-STRING

<Dial-command> := <Dial-operator> {<Dial-operator>}

<Dial-operator> := "0" | ... | "9" | "*" | "#" | ";" | ":" | "," | "T" | "t" | "P" | "p" | "W" | "w" | "@"

<Subaddress> := <Subaddress-parameter>
 -- *voir 11.2.2.7*

<Alias> := STRING

<Cover-path> := PATH
 -- *Trajet vers un fichier contenant une page de couverture pour un*
 -- *destinataire spécifique*

<Cover-conv> := <Convert-id-parameter>
 -- Spécifie le format de transfert du fichier page de couverture

<Cover-type> := <Type-id-parameter>
 -- Spécifie le format de transmission de la page de couverture

<Nopgrbk> := <Boolean-parameter>
 -- lorsqu'il est positionné à vrai ("Yes"), il n'y a pas de rupture de page
 -- entre la page de couverture et le document principal

11.2.2.5 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre File. Le paramètre File lui-même est codé comme un trajet PATH pointant sur le fichier transféré.

Syntaxe:

<File-of-filespec> := PATH
 -- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
 -- <Filespec-parameter>

<Filespec-parameter> := <File-parameter> "," <File-conv> ["," <File-type>]

<File-parameter> := PATH
 -- Trajet vers le fichier à transférer

<File-conv> := <Convert-id-parameter>
 -- Spécifie le format de transfert du fichier

<File-type> := <Type-id-parameter>
 -- Spécifie le format de transmission du fichier

11.2.2.6 Paramètre G3-speed

Le paramètre G3-speed est codé comme une chaîne NUMERIC-STRING prenant l'une des valeurs suivantes:

"2400"	2400 bit/s
"4800"	4800 bit/s
"7200"	7200 bit/s
"9600"	9600 bit/s
"12200"	12200 bit/s
"14400"	14400 bit/s

Syntaxe:

<G3-speed-parameter> := "2400" | "4800" | "7200" | "9600" | "12200" | "14400"

11.2.2.7 Paramètre Sub-address

Le paramètre Sub-address est codé comme une chaîne STRING qui représente une sous-adresse. Définie conformément à la Recommandation UIT-T T.30 (1994), la chaîne STRING est restreinte aux chiffres "0" à "9" et aux caractères "+" et "-". Le caractère "_" (soulignement) doit être transposé en un caractère espace IRA (20_{HEX}) par l'application de communication. La longueur du paramètre Sub-address est limitée à 20 octets.

Syntaxe:

<Sub-address-parameter> := <Subaddress-digit> {<Subaddress-digit>}

<Subaddress-digit> := "0" | ... | "9" | "*" | "#"
 -- La longueur du paramètre <Sub-address-parameter> est limitée à 20 octets.

11.2.2.8 Paramètre Resolution

Le paramètre Resolution est codé comme une chaîne NUMERIC-STRING prenant l'une des valeurs suivantes:

"0"	8 pel/mm horizontal, 3,85 ligne/mm vertical (98 dpi vertical)
"1"	8 pel/mm horizontal, 7,7 ligne/mm vertical (196 dpi vertical)
"2"	200 dpi horizontal, 200 dpi vertical (200 dpi)
"3"	300 dpi horizontal, 300 dpi vertical (300 dpi)
"4"	400 dpi horizontal, 300 dpi vertical (400 dpi)
"5"	8 pel/mm horizontal, 15,4 ligne/mm vertical (392 dpi vertical)
"6"	16 pel/mm horizontal, 15,4 ligne/mm vertical (392 dpi vertical)

Syntaxe:

<Resolution-parameter> := "0" | ... | "6"

11.2.2.9 Paramètre Password

Le paramètre Password est codé comme une chaîne STRING. Définie conformément à la Recommandation UIT-T T.30 (1994), la chaîne STRING est restreinte aux chiffres "0" à "9" et aux caractères "+" et "-". Le caractère "_" (soulignement) doit être transposé en un caractère espace IRA (20_{HEX}) par l'application de communication.

Syntaxe:

<Password-parameter> := <Password-digit> {<Password-digit>}

<Password-digit> := "0" | ... | "9" | "*" | "#"

-- La longueur du paramètre <Password-parameter> est limitée à 20 octets.

11.2.2.10 Paramètre Poll-select

Le paramètre Poll-select est codé comme une chaîne STRING. Définie conformément à la Recommandation UIT-T T.30 (1994), la chaîne STRING est restreinte aux chiffres "0" à "9" et aux caractères "+" et "-". Le caractère "_" (soulignement) doit être transposé en un caractère espace IRA (20_{HEX}) par l'application de communication.

Syntaxe:

<Poll-select-parameter> := <Poll-select-digit> {<Poll-select-digit>}

<Poll-select-digit> := "0" | ... | "9" | "*" | "#"

-- La longueur du paramètre <Poll-select-parameter> est limitée à 20 octets.

11.3 Fonctionnalités additionnelles

11.3.1 Fonctions: Send et SendAck

Voir les Tableaux 68 et 69.

11.3.2 Fonction: Receive

Voir le Tableau 70.

11.3.3 Fonction: Poll

Cette extension consiste à donner à l'application locale la possibilité de récupérer une série de documents envoyés par un télécopieur distant.

Par exemple, si l'utilisateur "Pierre" envoie un document au télécopieur distant de "Paul" et si "Paul" a envoyé des documents que "Pierre" doit récupérer, l'extension "Poll" (interrogation) permet à "Pierre" de recevoir les documents adressés par "Paul" dans la même session de communication.

La description <ExtendTDD>, associée à l'extension <PollExtension> est utilisée pour récupérer en particulier les documents adressés par un dispositif distant (voir le Tableau 71).

TABLEAU 68/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour la télécopie du groupe 3

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<G3Speed>	B	o	I	G3SPEED	<G3-speed-parameter>	La plus grande vitesse de modulation disponible	Rapidité de modulation, ce champ est vierge lorsque l'application de communication ne peut retourner la rapidité de modulation
<GenCil>	+	o	I	GENCIL	<Boolean-parameter>	"Yes"	Demande à l'application de communication de produire une ligne CIL dans le fichier transmis
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Impose une résolution plus élevée lorsque sa valeur est supérieure à "0". Voir aussi 11.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	-	Sous-adresse de l'expéditeur
<UseEcm>	+	o	I	USEECM	<Boolean-parameter>	"Yes"	Indique si le mode correction d'erreur doit être utilisé
Envoi à un destinataire							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	-	Spécifie le numéro d'appel d'un destinataire
<DoPoll>	+	o	I	DO POLL	<Boolean-parameter>	"No"	Si sa valeur est "Yes", le récepteur est interrogé pour l'envoi d'un document
<PollPassword>	+	o	I	PASSWORD	<Password-parameter>	-	
<PollSelector>	+	o	I	SELECT	<Poll-select-parameter>	-	
Envoi à un ou à plusieurs destinataires							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	-	Spécifie une liste de destinataires
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	-	Fichier unique sortant à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Numéro de la première page à envoyer réellement, ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Numéro de la dernière page à envoyer réellement, ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Spécifie une liste de fichiers, utilise une syntaxe spéciale

TABLEAU 69/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour la télécopie du groupe 3

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Recipient>	B	m	I/O	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<G3Speed>	B	o	I/O	G3SPEED	<G3-speed-parameter>	Plus grande vitesse disponible	Rapidité de modulation, ce champ est vierge lorsque l'application de communication ne peut retourner la vitesse de modulation
<GenCil>	+	o	I	GENCIL	<Boolean-parameter>	"Yes"	Demande à l'application de communication de produire une ligne CIL dans le fichier transmis
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Impose une résolution plus élevée lorsque sa valeur est supérieure à "0". Voir aussi 11.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
<UseEcm>	+	o	I/O	USEECM	<Boolean-parameter>	"Yes"	Indique si le mode de correction d'erreur doit être utilisé
<DoPoll>	+	o	I	DOPOLL	<Boolean-parameter>	"No"	Si sa valeur est "Yes", le récepteur est interrogé pour l'envoi d'un document
<PollPassword>	+	o	I	PASSWORD	<Password-parameter>	–	
<PollSelector>	+	o	I	SELECT	<Poll-select-parameter>	–	
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique sortant à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement, ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement, ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe spéciale

TABLEAU 70/T.611

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour la télécopie du groupe 3

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<CvFax3>	B	o	I	CVFAX3	<Convert-id-parameter>	"TIFF"	Format de transfert souhaité par l'application locale
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Spécifie le numéro de téléphone de l'expéditeur
<G3Speed>	+	o	O	G3SPEED	<G3-speed-parameter>	–	Rapidité de modulation. Ce champ est vide lorsque l'application de communication ne peut pas renvoyer la rapidité de modulation
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Sous-adresse du destinataire spécifiée dans Request, utilisé comme sélecteur pour l'interrogation
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance par l'application locale sur demande. En cas de réception d'un seul fichier, le nom doit être conservé par l'application locale. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe spéciale

11.4 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Lorsqu'une application de communication prend en charge le service de télécopie du groupe 3, elle doit spécifier les identificateurs Type-ID pris en charge (formats de transmission des documents) dans le descripteur CA-Descriptor (voir 9.5).

La prise en charge du descripteur TDD Poll impose un choix approprié du mot clé EXTEND du descripteur CA-Descriptor (voir le Tableau 72).

TABLEAU 71/T.611

Éléments syntaxiques pour l'extension <PollExtension> dans le cas de la télécopie du groupe 3

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<PollSubFunction>	B	m	I	SUBFUNC	"Poll"	–	L'application de communication doit produire une réponse
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<PollPassword>	B	m	I	PASSWORD	<Password-parameter>	–	
<PollSelector>	B	o	I	SELECT	<Poll-select-parameter>		
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	L'application de communication doit réellement traiter la demande à la date spécifiée
<ComId>	B	o	O	COMID	<Com-id-parameter>	–	Identification de la communication (déterminée par l'application de communication)

TABLEAU 72/T.611

Valeurs additionnelles du descripteur CA-Descriptor pour la télécopie du groupe 3

Mot clé	Paramètre	L'application de communication déclare que
FX3	"STD"	le service de télécopie de base (MH) est assuré
FX3	"BTM"	le mode transparent de base est assuré
FX3	"DTM"	le mode document transparent est assuré
FX3	"BFT"	le transfert de fichiers binaires est assuré
FX3	"EDI"	le transfert électronique de données est assuré
EXTEND	"Poll"	la fonction Poll est prise en charge

12 Service: Télécopie du groupe 4

12.1 Eléments syntaxiques propres au service

<ServiceDependentKeywordsSend> :=
 (<Recipient> | <RecipientSpec>)
 (((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [From]
 [To]) | <DocumentSpec>) [<SubAddress>] [<HighRes>]

<ServiceDependentKeywordsSendAck> :=
 <Recipient>
 (((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]
 [From] [To]) | <DocumentSpec>) [<SubAddress>] [<HighRes>]

<ServiceDependentKeywordsReceive> :=
 [<Originator>]
 (((<Document> <Convert> <Type> [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]
 [<FirstPg>]) | <DocumentSpec>) [<CvFax4>] [<SubAddress>]

Voir le Tableau 73.

TABLEAU 73/T.611

Eléments syntaxiques supplémentaires pour la télécopie du groupe 4

Elément syntaxique	Objet
<Cil>	Spécifie le contenu de la ligne d'identification de l'appel (Call Identification Line) définie dans la Recommandation UIT-T F.200. Voir aussi G.1
<Convert>	Spécifie le format de transfert à utiliser
<CvFax4>	Spécifie le format de transfert souhaité pour les fichiers reçus
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception
<FirstPg>	Spécifie le numéro de la page reçue en premier
<From>	Spécifie le numéro de page où la transmission doit commencer
<HighRes>	Indique la résolution
<Name>	Assigne un nom au document à envoyer ou indique le nom du document reçu tel que défini dans la Recommandation UIT-T T.571
<Originator>	Spécifie l'adresse de communication de l'expéditeur
<Prolog>	Trajet vers le document "prologue". Ce document, codé conformément à la Recommandation UIT-T T.61, précède le document principal à envoyer
<Recipient>, <RecipientSpec>	Spécifie la ou les adresses de communication du ou des destinataires
<SubAddress>	Spécifie la sous-adresse de l'expéditeur
<To>	Spécifie le numéro de page où la transmission doit s'arrêter
<Type>	Spécifie le format de transmission utilisé
<UserInfo>	Spécifie un commentaire associé au document à envoyer ou à recevoir. Ce commentaire est transmis par les services de télécommunication

12.2 Codage de type texte

12.2.1 Correspondance des mots clés

Voir le Tableau 74.

TABLEAU 74/T.611

Codage de type texte des éléments syntaxiques additionnels pour la télécopie du groupe 4 (Le signe ↵ est le caractère "ligne suivante")

Élément syntaxique	Couple mot clé/paramètre
<Cil>	"CIL" ":" <Cil-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvFax4>	"CVFAX4" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<FirstPg>	"FIRSTPG" ":" NUMERIC-STRING ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<HighRes>	"HIGHRES" ":" <Resolution-parameter> ↵
<Name>	"NAME" ":" STRING (SIZE(1..12)) ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Prolog>	"PROLOG" ":" <Path-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UserInfo>	"USERINFO" ":" STRING (SIZE(1..12)) ↵

12.2.2 Codage des paramètres

Voir aussi 6.4.4 pour le codage des paramètres non dépendants du service utilisés.

12.2.2.1 Paramètre Service-id

Le paramètre Service-id est codé comme une chaîne STRING dont la valeur est constante et égale à "FX4".

Syntaxe:

<Service-id-parameter> := "FX4"

12.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"STD"	Service de télécopie de base du groupe 4 (MR)
"DTM"	Transfert de fichier télématique (TFT) du service de télécopie du groupe 4: mode document transparent
"EDI"	Transfert de fichier télématique (TFT) du service de télécopie du groupe 4: Edifact
"BFT"	Service de télécopie du groupe 4: transfert de fichier binaire (BFT)

NOTE – Le format de transmission STD (= code Read modifié, MR) est défini dans la Recommandation UIT-T T.6, les formats TFT DTM et EDI sont définis dans la Recommandation UIT-T T.571 et le format de transmission BFT est défini dans la Recommandation UIT-T T.434.

Syntaxe:

<Type-id-parameter> := "STD" | "DTM" | "BFT" | "EDI"

12.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"ASCII"	ASCII étendu APPLI/COM (pour les fichiers sortants seulement)
"ASCII437"	ASCII étendu APPLI/COM (pour les fichiers sortants seulement)
"T.50"	ASCII étendu APPLI/COM (pour les fichiers sortants seulement)
"T.61"	Format de transfert APPLI/COM T.61
"TIFF"	Format de transfert TIFF défini au 8.3
"TIFF2"	(pour les fichiers entrants seulement) Classe 2 APPLI/COM TIFF Class 2 spécifiée dans 8.3
"TIFF4"	(pour les fichiers entrants seulement) Classe 4 APPLI/COM TIFF Class 4 spécifiée dans 8.3
"VOID"	Pas de conversion à effectuer

NOTE – Si une application locale réclame un document de télécopie qui a été reçu, il est possible:

- d'obtenir le document dans le format TIFF de classe 2 APPLI/COM en spécifiant TIFF2;
- d'obtenir le document dans le format TIFF de classe 4 APPLI/COM en spécifiant TIFF4;

pour autant que l'application de communication soit capable de produire ce format TIFF particulier. Si seul le format TIFF est spécifié, le document sera remis dans la classe TIFF APPLI/COM par défaut (classe 1). Toutefois, dans le sens émission, il suffit uniquement de spécifier le format TIFF (sans extension de numéro), car l'information de compression est contenue dans le format de transfert lui-même.

L'utilisation du paramètre Convert-id dépend aussi de l'identificateur Type-ID choisi (voir le Tableau 75).

TABLEAU 75/T.611

Valeurs autorisées du paramètre Convert-id qui dépendent du paramètre Type-id et du sens du trafic

Type-id	Convert-id pour le trafic sortant	Convert-id pour le trafic entrant
STD	ASCII, ASCII ^{a)} , T.50, TIFF	TIFF, TIFF _x ^{b)}
DTM, BFT, EDI	VOID	
<p>^{a)} xxx est un paramètre code-page, déclaré dans l'ICE, ainsi ce paramètre est ASCII437 si le code-page 437 a été déclaré.</p> <p>^{b)} x désigne la classe de format TIFF à lire, qui peut prendre les valeurs 2 ou 4.</p>		

Syntaxe:

```
<Convert-id-parameter> := <Convert-std> | <Convert-bin> | <Convert-txt>
<Convert-std> := <Convert-std-in> | <Convert-std-out>
<Convert-std-in> := "TIFF" | "TIFF2" | "TIFF4"
-- documents entrants
<Convert-std-out> := <Ascii> | "T.50" | "TIFF"
-- documents sortants
<Convert-txt> := <Ascii> | "T.50" | "T.61"
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
<Convert-bin> := "VOID"
```

12.2.2.4 Paramètres File-of-addrspec et Address

Le paramètre File-of-addrspec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Addrspec, qui contiennent le paramètre Address.

Le paramètre Address est codé comme une chaîne STRING. La chaîne STRING se compose de l'identificateur du terminal du destinataire tel que défini dans la Recommandation UIT-T F.184. La partie mnémotechnique de cet identificateur peut être omise ou spécifiée sous la forme d'un point d'interrogation ("?"), auquel cas l'adresse de l'abonné n'est pas vérifiée. Voir aussi G.2.

On peut aussi donner un nom de substitution à la place du numéro de téléphone, mais il faut alors que ce nom soit précédé du caractère "&". On suppose que l'application de communication sait décoder le nom de substitution spécifié. Cette caractéristique est généralement utilisée lorsque l'application de communication met en œuvre un annuaire téléphonique.

Syntaxe:

```
<File-of-addrspec> := PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Addrspec-parameter>
<Addrspec-parameter> := <Address-parameter>
["," <Cover-path> "," <Cover-conv> ["," [<Cover-type>] ["," <Nopgbrk>]]]
<Address-parameter> := <Terminal-id> | ("&" <Alias>)
<Terminal-id> := STRING
-- La chaîne doit présenter l'identificateur terminal-id tel que défini dans la
-- Recommandation UIT-T F.184
<Alias> := STRING
<Cover-path> := PATH
-- Trajet vers un fichier contenant une page de couverture pour
-- un destinataire spécifique
<Cover-conv> := <Convert-id-parameter>
-- Spécifie le format de transfert du fichier pages de couverture
<Cover-type> := <Type-id-parameter>
-- Spécifie le format de transmission de la page de couverture
<Nopgbrk> := <Boolean-parameter>
-- Lorsqu'il est positionné à vrai ("Yes"), il n'y a pas de rupture de page
-- entre la page de couverture et le document principal
```

12.2.2.5 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH renvoyant à un fichier contenant les paramètres Filespec, qui contiennent à leur tour le paramètre File. Le paramètre File lui-même est codé comme un trajet PATH pointant sur le fichier transféré.

Syntaxe:

```
<File-of-filespec> :=          PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Filespec-parameter>

<Filespec-parameter> :=      <File-parameter> "," <File-conv> ["," [<File-type>]
["," <Send-parameter> | <SendAck-parameter> | <Receive-parameter>] ]

<Send-parameter> :=          [<File-name>] ["," <File-userinfo>]

<SendAck-parameter> :=       [<File-name>] ["," [<File-userinfo>] ["," <File-cil>] ]

<Receive-parameter> :=       [<File-name>] ["," [<File-userinfo>] ["," <File-cil>] ]

<File-parameter> :=          PATH
-- Trajet vers le fichier transféré

<File-conv> :=                <Convert-id-parameter>
-- Spécifie le format de transfert du fichier

<File-type> :=                <Type-id-parameter>
-- Spécifie le format de transmission du fichier

<File-name> :=                STRING (SIZE(1..12))
-- Spécifie le nom du fichier à transmettre

<File-userinfo> :=            STRING (SIZE(1..12))
-- Spécifie l'information complémentaire d'utilisateur à utilisateur à transmettre

<File-cil> :=                 Cil-parameter
-- Voir 12.2.2.6
```

12.2.2.6 Paramètre Cil

Ce paramètre est une chaîne de 72 caractères définie dans la Recommandation UIT-T F.200, composée comme suit (voir aussi G.1):

Identificateur du terminal du destinataire	Longueur: 24 caractères
Séparateur	Caractère "/" (2F _{HEX})
Identificateur du terminal de l'expéditeur	Longueur: 24 caractères
Séparateur	Caractère "/" (2F _{HEX})
Heure-date locale pour l'expéditeur	Longueur: 14 caractères, format: "YY-MM-DD-HH:MM"
Séparateur	Caractère "/" (2F _{HEX})
Information de référence	Longueur: 7 caractères

Syntaxe:

```
<Cil-parameter> :=           <Recipient-tid> "/" <Originator-tid> "/" <Local-time> "/" <Ref-info>

<Recipient-tid> :=           STRING (SIZE(24))
-- La chaîne contient l'identificateur terminal-id tel que défini dans la
-- Recommandation UIT-T F.184

<Originator-tid> :=          STRING (SIZE(24))
-- La chaîne contient l'identificateur terminal-id tel que défini dans la
-- Recommandation UIT-T F.184

<Local-time> :=              <Date-time-parameter>

<Ref-info> :=                 STRING (SIZE(7))
```

12.2.2.7 Paramètre Sub-address

Le paramètre Sub-address est codé comme une chaîne NUMERIC-STRING représentant une sous-adresse.

Syntaxe:

<Sub-address-parameter> := NUMERIC-STRING (SIZE(1..4))

12.2.2.8 Paramètre Resolution

Le paramètre Resolution est codé comme une chaîne NUMERIC-STRING qui prend l'une des valeurs suivantes:

"0"	200 dpi
"1"	240 dpi
"2"	300 dpi
"3"	400 dpi
"4"	400 dpi

Syntaxe:

<Resolution-parameter> := "0" | ... | "4"

12.3 Fonctionnalités additionnelles

12.3.1 Fonctions: Send et SendAck

Voir les Tableaux 76 et 77.

12.3.2 Fonction: Receive

Voir le Tableau 78.

12.4 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Lorsqu'une application de communication prend en charge le service de télécopie du groupe 4, elle doit spécifier l'identificateur Type-id pris en charge (formats de transmission des documents) dans le descripteur CA-Descriptor (voir 9.5). Le Tableau 79 montre les valeurs que le descripteur CA peut prendre.

13 Service: Télétex

13.1 Eléments syntaxiques propres au service

<ServiceDependentKeywordsSend> :=
(<Recipient> | <RecipientSpec>)
(((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [From]
[To] [T61Options]) | <DocumentSpec>) [<SubAddress>]

<ServiceDependentKeywordsSendAck> :=
<Recipient>
(((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]
[From] [To] [T61Options]) | <DocumentSpec>) [<SubAddress>]

<ServiceDependentKeywordsReceive> :=
[<Originator>]
(((<Document> <Convert> <Type> [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]
[<FirstPg>]) | <DocumentSpec>) [<CvTtx>] [<SubAddress>]

Voir le Tableau 80.

TABLEAU 76/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour la télécopie du groupe 4

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Impose une résolution plus élevée lorsque sa valeur est supérieure à "0". Voir aussi 12.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi à un destinataire							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
Envoi à un ou à plusieurs destinataires							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Spécifie une liste de destinataires
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Assigne un nom au fichier à envoyer. Il comporte 12 caractères au maximum. Il correspond au paramètre nom de fichier du transfert TFT (voir Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Assigne un commentaire d'utilisateur au document. Ce commentaire est transmis avec le document. Il comporte 12 caractères au maximum
<Prolog>	+	o	I	PROLOG	PATH	–	Nom de fichier du document de commande (trajet complet)
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement; ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement; ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 77/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour la télécopie du groupe 4

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Impose une résolution plus élevée lorsque sa valeur est supérieure à "0". Voir aussi 12.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi à un destinataire							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<Cil>	B	o	O	CIL	<Cil-parameter>	–	Ligne d'identification de l'appel créée par l'application de communication (voir la Rec. UIT-T F.200)
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Assigne un nom au fichier à envoyer. Il comporte 12 caractères au maximum. Il correspond au paramètre nom de fichier du transfert TFT (voir Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Assigne un commentaire d'utilisateur au document. Ce commentaire est transmis avec le document. Il comporte 12 caractères au maximum
<Prolog>	+	o	I	PROLOG	PATH	–	Nom de fichier du document de commande (trajet complet)
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement; ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement; ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 78/T.611

Fonctions supplémentaires de la fonction <ReceiveTDD> pour la télécopie du groupe 4

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Spécifie le numéro de téléphone de l'expéditeur
<CvFax4>	B	o	I	CVFAX4	<Convert-id-parameter>	"TIFF"	Format de transfert souhaité par l'application locale
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Sous-adresse du destinataire spécifiée dans Request, utilisé comme sélecteur pour l'interrogation
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance par l'application locale sur demande. En cas de réception d'un seul fichier, le nom doit être conservé par l'application de communication. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
<Cil>	B	o	O	CIL	<Cil-parameter>	–	Ligne d'identification de l'appel produit par l'application de communication (voir la Rec. UIT-T F.200)
<Name>	+	o	O	NAME	STRING (SIZE(1..12))	–	Renvoie le nom du fichier entrant (12 caractères au maximum)
<UserInfo>	+	o	O	USERINFO	STRING (SIZE(1..12))	–	Assigne un commentaire d'utilisateur au document. Ce commentaire est transmis avec le document. Il comporte 12 caractères au maximum
<FirstPg>	+	o	O	FIRSTPG	NUMERIC-STRING	–	Numéro de la première page
<Prolog>	+	o	I	PROLOG	PATH	–	Nom de fichier du document de commande
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 79/T.611

Valeurs additionnelles du descripteur CA-Descriptor pour la télécopie du groupe 4

Mot clé	Paramètre	L'application de communication déclare que
FX4	"STD"	le service de télécopie de base est assuré
FX4	"OPD"	la transmission du document d'opérateur est assurée
FX4	"MD"	la transmission du document de supervision est assurée
FX4	"CTL"	la transmission du document de commande est assurée
FX4	"DTM"	le mode document transparent est assuré
FX4	"BFT"	le transfert de fichiers binaires est assuré
FX4	"EDI"	le transfert électronique de données est assuré

TABLEAU 80/T.611

Eléments syntaxiques supplémentaires pour le service télétex

Elément syntaxique	Objet
<Cil>	Spécifie le contenu de la ligne d'identification de l'appel (Call Identification Line) définie dans la Recommandation UIT-T F.200. Voir aussi G.1
<Convert>	Spécifie le format de transfert à utiliser
<CvTtx>	Spécifie le format de transfert souhaité pour les fichiers reçus
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception
<FirstPg>	Spécifie le numéro de la page reçue en premier
<From>	Spécifie le numéro de page où la transmission doit commencer
<Name>	Assigne un nom au document à envoyer ou indique le nom du document reçu tel que défini dans la Recommandation UIT-T T.571
<Originator>	Spécifie l'adresse de communication de l'expéditeur
<Prolog>	Trajet vers le document "prologue"
<Recipient>, <RecipientSpec>	Spécifie la ou les adresses de communication du ou des destinataires
<SubAddress>	Spécifie la sous-adresse de l'expéditeur
<T61Options>	Indique les options spécifiques pour le codage T.61
<To>	Spécifie le numéro de page où la transmission doit s'arrêter
<Type>	Spécifie le format de transmission utilisé
<UserInfo>	Spécifie un commentaire associé au document à envoyer ou à recevoir. Ce commentaire est transmis par les services de télécommunication

13.2 Codage de type texte

13.2.1 Correspondance des mots clés

Voir le Tableau 81.

TABLEAU 81/T.611

Codage de type texte des éléments syntaxiques additionnels pour le service télétext
(Le signe ↵ est le caractère "ligne suivante")

Elément de syntaxe	Couple mot clé/paramètre
<Cil>	"CIL" ":" <Cil-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvTtx>	"CVTTX" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<FirstPg>	"FIRSTPG" ":" NUMERIC-STRING ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<Name>	"NAME" ":" (SIZE (1..12)) ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Prolog>	"PROLOG" ":" <Path-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<T61Options>	"T61OPTS" ":" <T61-options-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UserInfo>	"USERINFO" ":" STRING (SIZE (1..12)) ↵

13.2.2 Codage des paramètres

Voir aussi 6.4.4 pour le codage des paramètres non dépendants du service utilisés.

13.2.2.1 Paramètre Service-id

Le paramètre Service-id est codé comme une chaîne STRING dont la valeur est constante et égale à "TTX".

Syntaxe:

<Service-id-parameter> := "TTX"

13.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"STD"	Service télétexte de base (T.61)
"OPD"	Service télétexte de base: document d'opérateur
"MD"	Service télétexte de base: document de supervision
"CTL"	Service télétexte de base: document de commande
"DTM"	Transfert de fichier télématique (TFT) du service télétexte: mode document transparent
"EDI"	Transfert de fichier télématique (TFT) du service télétexte: Edifact
"BFT"	Service télétexte: transfert de fichier binaire

NOTE – Le format de transmission STD est défini dans la Recommandation UIT-T T.61, les types de document OPD, MD et CTL sont définis dans la Recommandation UIT-T T.62 (Annexe E), les formats TFT DTM et EDI sont définis dans la Recommandation UIT-T T.571 et le format de transmission BFT est défini dans la Recommandation UIT-T T.434.

Syntaxe:

<Type-id-parameter> := "STD" | "OPD" | "MD" | "CTL" | "DTM" | "BFT" | "EDI"

13.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"ASCII"	ASCII étendu APPLI/COM
"ASCII437"	ASCII étendu APPLI/COM
"T.50"	ASCII normalisé APPLI/COM
"T.61"	Codage du service télétexte de base défini dans la Recommandation UIT-T T.61
"VOID"	Pas de conversion à effectuer

L'utilisation du paramètre Convert-id dépend aussi du paramètre Type-id choisi (voir le Tableau 82).

TABLEAU 82/T.611

Valeurs autorisées du paramètre Convert-id qui dépendent du paramètre Type-id

Paramètre Type-id	Valeurs autorisées du paramètre Convert-id pour le trafic entrant et le trafic sortant
STD, OPD, MD, CTL	ASCII, ASCII _{xxx} ^{a)} , T.50, T.61
DTM, BFT, EDI	VOID
a) xxx est un code-page, déclaré dans l'ICE; par exemple, ce paramètre est ASCII437 si le code-page 437 a été déclaré.	

Syntaxe:

<Convert-id-parameter> := <Convert-txt> | <Convert-bin>
<Convert-txt> := <Ascii> | "T.50" | "T.61"
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
<Convert-bin> := "VOID"

13.2.2.4 Paramètres File-of-addrspec et Address

Le paramètre File-of-addrspec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Addrsec, qui contiennent le paramètre Address.

Le paramètre Address est codé comme une chaîne STRING. La chaîne STRING se compose de l'identificateur du terminal du destinataire tel que défini dans la Recommandation UIT-T F.200. La partie mnémonique de cet identificateur peut être omise ou spécifiée sous la forme d'un point d'interrogation ("?"), auquel cas l'adresse de l'abonné n'est pas vérifiée. Voir aussi G.2.

De même, la présence d'un numéro d'appel sous la forme d'un numéro brut à composer peut être indiquée en mettant un point d'exclamation ("!") avant le premier chiffre. Ainsi, l'équipement appelant doit composer sans interprétation les chiffres qui suivent le point d'exclamation.

On peut aussi donner un nom de substitution à la place de l'identificateur de terminal, mais il faut alors que ce nom soit précédé du caractère "&". On suppose que l'application de communication sait décoder le nom de substitution spécifié.

Syntaxe:

```
<File-of-addrsec> :=          PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Addrsec-parameter>

<Addrsec-parameter> :=      <Address-parameter>

<Address-parameter> :=      <Terminal-id> | ("!" <Dial-sequence>) | ("&" <Alias>)

<Terminal-id> :=            STRING
-- La chaîne doit présenter l'identificateur terminal-id tel que défini dans la
-- Recommandation UIT-T F.200

<Dial-sequence> :=         NUMERIC-STRING

<Alias> :=                  STRING
```

13.2.2.5 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre File. Le paramètre File lui-même est codé comme un trajet PATH pointant sur le fichier transféré.

Syntaxe:

```
<File-of-filespec> :=        PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Filespec-parameter>

<Filespec-parameter> :=      <File-parameter> ", " <File-conv> [", " [<File-type>]
[" , " <Send-parameter> | <SendAck-parameter> | <Receive-parameter> ] ]

<Send-parameter> :=          [<File-name>] [", " <File-userinfo>]

<SendAck-parameter> :=      [<File-name>] [", " [<File-userinfo>] [", " <File-cil>] ]

<Receive-parameter> :=      [<File-name>] [", " [<File-userinfo>] [", " <File-cil>] ]

<File-parameter> :=         PATH
-- Trajet vers le fichier transféré

<File-conv> :=               <Convert-id-parameter>
-- Spécifie le format de transfert du fichier

<File-type> :=               <Type-id-parameter>
-- Spécifie le format de transmission du fichier

<File-name> :=               STRING (SIZE(1..12))
-- Spécifie le nom du fichier à transmettre
```

<File-userinfo> := STRING (SIZE(1..12))
 -- Spécifie l'information complémentaire d'utilisateur à utilisateur à transmettre

<File-cil> := Cil-parameter
 -- Voir 13.2.2.6

13.2.2.6 Paramètre Cil

Ce paramètre est une chaîne de 72 caractères définie dans la Recommandation UIT-T F.200, composée comme suit (voir aussi G.1):

Identificateur du terminal du destinataire	Longueur: 24 caractères
Séparateur	Caractère "/" (2F _{HEX})
Identificateur du terminal de l'expéditeur	Longueur: 24 caractères
Séparateur	Caractère "/" (2F _{HEX})
Heure-date locale pour l'expéditeur	Longueur: 14 caractères, format: "YY-MM-DD-HH:MM"
Séparateur	Caractère "/" (2F _{HEX})
Information de référence	Longueur: 7 caractères

Syntaxe:

<Cil-parameter> := <Recipient-tid> "/" <Originator-tid> "/" <Local-time> "/" <Ref-info>

<Recipient-tid> := STRING (SIZE(24))
 -- La chaîne contient l'identificateur terminal-id tel que défini dans la
 -- Recommandation UIT-T F.184

<Originator-tid> := STRING (SIZE(24))
 -- La chaîne contient l'identificateur terminal-id tel que défini dans la
 -- Recommandation UIT-T F.184

<Local-time> := <Date-time-parameter>

<Ref-info> := STRING (SIZE(7))

13.2.2.7 Paramètre Sub-address

Le paramètre Sub-address est codé comme une chaîne NUMERIC-STRING représentant une sous-adresse.

Syntaxe:

<Sub-address-parameter> := NUMERIC-STRING (SIZE(1..4))

13.2.2.8 Paramètre T61-options

Appelle un complément d'étude.

Syntaxe:

<T61-options-parameter> := STRING

13.3 Fonctionnalités additionnelles

13.3.1 Fonctions: Send et SendAck

Voir les Tableaux 83 et 84.

13.3.2 Fonction: Receive

Voir le Tableau 85.

TABLEAU 83/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour le service télétext

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi à un destinataire							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
Envoi à un ou à plusieurs destinataires							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Spécifie une liste de destinataires.
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Paramètre nom de fichier dans le transfert TFT (T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Information d'utilisateur transmise avec le document
<Prolog>	+	o	I	PROLOG	PATH	–	Trajet vers le document de contrôle (T.62)
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement; ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement; ne s'applique qu'aux fichiers textuels
<T61Options>	+	o	I	T61OPTS	<T61-options-parameter>	–	
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe spéciale

TABLEAU 84/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour le service télétext

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique sortant à transmettre, remis dans le format spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<Cil>	B	o	O	CIL	<Cil-parameter>	–	Ligne d'identification de l'appel produite par l'application de communication (voir Rec. UIT-T F.200)
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Assigne un nom aux fichiers à envoyer. Il correspond au paramètre nom de fichier du transfert TFT (voir Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Assigne un commentaire d'utilisateur au document. Ce commentaire est transmis avec le document
<Prolog>	+	o	I	PROLOG	PATH	–	Nom de fichier du document de commande (trajet complet)
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement; ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement; ne s'applique qu'aux fichiers textuels
<T61Options>	+	o	I	T61OPTS	<T61-options-parameter>	–	
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 85/T.611

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour le service télétext

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Spécifie le numéro de téléphone de l'expéditeur
<CvTx>	B	o	I	CVTTX	<Convert-id-parameter>	"T.61"	Format de transfert souhaité par l'application locale
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Sous-adresse du destinataire; lorsqu'elle est spécifiée dans la demande, elle sert de sélecteur de recherche
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance par l'application locale. En cas de réception d'un seul fichier, le trajet doit être conservé par l'application de communication. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
<Cil>	B	o	O	CIL	<Cil-parameter>	–	Ligne d'identification de l'appel produite par l'application de communication (voir la Rec. UIT-T F.200)
<Name>	+	o	O	NAME	STRING (SIZE(1..12))	–	Renvoie le nom du fichier entrant (12 caractères au maximum)
<UserInfo>	+	o	O	USERINFO	STRING (SIZE(1..12))	–	Assigne un commentaire d'utilisateur au document. Ce commentaire est transmis avec le document. Il comporte 12 caractères au maximum
<FirstPg>	+	o	O	FIRSTPG	NUMERIC-STRING	–	Numéro de la première page reçue
<Prolog>	+	o	I	PROLOG	PATH	–	Nom de fichier du document de commande
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers, utilise une syntaxe particulière

13.4 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Lorsqu'une application de communication prend en charge le service télétext, elle doit spécifier l'identificateur Type-id pris en charge (formats de transmission des documents) dans le descripteur CA-Descriptor (voir 9.5). Le Tableau 86 montre les valeurs que ce descripteur peut prendre.

TABLEAU 86/T.611

Valeurs supplémentaires du descripteur CA-Descriptor pour le service télételex

Mot clé	Paramètre	L'application de communication déclare que
TTX	"STD"	le service télételex de base (T.61) est assuré
TTX	"OPD"	la transmission du document d'opérateur est assurée
TTX	"MD"	la transmission du document de supervision est assurée
TTX	"CTL"	la transmission du document de commande est assurée
TTX	"DTM"	le mode document transparent est assuré
TTX	"BFT"	le transfert de fichiers binaires est assuré
TTX	"EDI"	le transfert électronique de données est assuré

14 Service: Télex par télételex

14.1 Eléments syntaxiques propres au service

<ServiceDependentKeywordsSend> :=
 (<Recipient> | <RecipientSpec>)
 ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)
 [<SubAddress>] [<Notify>]

<ServiceDependentKeywordsSendAck> :=
 <Recipient>
 ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)
 [<SubAddress>] [<Notify>]

<ServiceDependentKeywordsReceive> :=
 [<Originator>] ((<Document> <Convert> <Type>) | <DocumentSpec>)
 [<CvTx>] [<SubAddress>]

Voir le Tableau 87.

TABLEAU 87/T.611

Eléments syntaxiques supplémentaires du télex par télételex

Elément syntaxique	Objet
<Convert>	Spécifie le format de transfert à utiliser
<CvTx>	Spécifie le format de transfert souhaité pour les fichiers reçus
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception
<From>	Spécifie le numéro de page où la transmission doit commencer
<Notify>	Indique si la notification de remise est demandée
<Originator>	Spécifie l'adresse de communication de l'expéditeur
<Recipient>, <RecipientSpec>	Spécifie la ou les adresses de communication du ou des destinataires
<SubAddress>	Spécifie la sous-adresse de l'expéditeur
<To>	Spécifie le numéro de page où la transmission doit s'arrêter
<Type>	Spécifie le format de transmission utilisé

14.2 Codage de type texte

14.2.1 Correspondance des mots clés

Voir le Tableau 88.

TABLEAU 88/T.611

Codage de type texte des éléments syntaxiques additionnels pour le télex par télétex
(Le signe ↵ est le caractère "ligne suivante")

Elément syntaxique	Couple mot clé/paramètre
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvTx>	"CVTX" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<Notify>	"NOTIFY" ":" <Boolean-parameter> ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵

14.2.2 Codage des paramètres

Voir aussi 6.4.4 pour le codage des paramètres non dépendants du service utilisés.

14.2.2.1 Paramètre Service-id

Le paramètre Type-id est codé comme une chaîne STRING dont la valeur est constante et égale à "TX".

Syntaxe:

<Service-id-parameter> := "TX"

14.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING dont la valeur est constante et égale à "STD".

Syntaxe:

<Type-id-parameter> := "STD"

NOTE – Le format de transmission STD est défini dans la Recommandation UIT-T S.1.

14.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"ASCII"	ASCII étendu APPLI/COM
"ASCII437"	ASCII étendu APPLI/COM
"T.50"	ASCII normalisé APPLI/COM

Syntaxe:

```
<Convert-id-parameter> := <Ascii> | "T.50"
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
```

14.2.2.4 Paramètres File-of-addrspec et Address

Le paramètre File-of-addrspec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Addrsec, qui contiennent le paramètre Address.

Le paramètre Address est codé comme une chaîne STRING formant le numéro d'appel. Lorsqu'on doit procéder à une vérification de l'identification de l'abonné, l'indicatif du terminal récepteur précédé du caractère égal ("=") doit être introduit après le numéro d'appel.

On peut aussi donner un nom de substitution à la place de l'identificateur de terminal, mais il faut alors que ce nom soit précédé du caractère "&". On suppose que l'application de communication sait décoder le nom de substitution spécifié.

NOTE – Si l'on veut procéder à une vérification de l'identification de l'abonné, l'indicatif complet doit être composé après le signe "=" et pas uniquement la composante alphabétique du récepteur.

Syntaxe:

```
<File-of-addrsec> := PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Addrsec-parameter>

<Addrsec-parameter> := <Address-parameter>

<Address-parameter> := <Telex-address> | ("&" <Alias>)

<Telex-address> := STRING
-- La chaîne doit représenter une adresse télex valide

<Alias> := STRING
```

14.2.2.5 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre File. Le paramètre File lui-même est codé comme un trajet PATH pointant sur le fichier transféré.

Syntaxe:

```
<File-of-filespec> := PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Filespec-parameter>

<Filespec-parameter> := <File-parameter> "," <File-conv> ["," <File-type>]

<File-parameter> := PATH
-- Trajet vers le fichier transféré

<File-conv> := <Convert-id-parameter>
-- Spécifie le format de transfert du fichier

<File-type> := <Type-id-parameter>
-- Spécifie le format de transmission du fichier
```

14.2.2.6 Paramètre Sub-address

Le paramètre Sub-address est codé comme une chaîne NUMERIC-STRING qui représente une sous-adresse.

Syntaxe:

<Sub-address-parameter> := NUMERIC-STRING

14.3 Fonctionnalités additionnelles

14.3.1 Fonctions: Send et SendAck

Voir les Tableaux 89 et 90.

TABLEAU 89/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour le télex par télétex

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Notify>	+	o	I	NOTIFY	<Boolean-parameter>	"Yes"	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi à un destinataire							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
Envoi à un ou à plusieurs destinataires							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Spécifie une liste de destinataires
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement, ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement, ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 90/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour le télex par télételex

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<Notify>	+	o	I	NOTIFY	<Boolean-parameter>	"Yes"	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement, ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement, ne s'applique qu'aux fichiers textuels.
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

14.3.2 Fonction: Receive

Voir le Tableau 91.

14.4 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Lorsqu'une application de communication prend en charge le service télex, elle doit spécifier l'identificateur Type-id pris en charge (formats de transmission des documents) dans le descripteur CA-Descriptor (voir 9.5). Le Tableau 92 montre les valeurs que ce descripteur peut prendre.

TABLEAU 91/T.611

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour le télex par télétex

Elément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Spécifie le numéro de téléphone de l'expéditeur
<CvTx>	B	o	I	CVTX	<Convert-id-parameter>	"T.50"	Format de transfert souhaité par l'application locale
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Sous-adresse du destinataire; lorsqu'elle est spécifiée dans la demande, elle sert de sélecteur de recherche
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance par l'application locale. En cas de réception d'un seul fichier, le trajet doit être conservé par l'application de communication. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 92/T.611

Valeurs supplémentaires du descripteur CA-Descriptor pour le télex par télétex

Mot clé	Paramètre	L'application de communication déclare que
TX	"STD"	le service de base est assuré

15 Service: Téléx

15.1 Eléments syntaxiques propres au service

<ServiceDependentKeywordsSend> :=
(<Recipient> | <RecipientSpec>)
(((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)
[<SubAddress>]

<ServiceDependentKeywordsSendAck> :=
<Recipient>
(((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)
[<SubAddress>]

<ServiceDependentKeywordsReceive> :=
[<Originator>] ((<Document> <Convert> <Type>) | <DocumentSpec>)
[<CvTlx>] [<SubAddress>]

Voir le Tableau 93.

TABLEAU 93/T.611

Eléments syntaxiques supplémentaires pour le service téléx

Elément syntaxique	Objet
<Convert>	Spécifie le format de transfert à utiliser
<CvTlx>	Spécifie le format de transfert souhaité pour les fichiers reçus
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception
<From>	Spécifie le numéro de page où la transmission doit commencer
<Originator>	Spécifie l'adresse de communication de l'expéditeur
<Recipient>, <RecipientSpec>	Spécifie la ou les adresses de communication du ou des destinataires
<SubAddress>	Spécifie la sous-adresse de l'expéditeur
<To>	Spécifie le numéro de page où la transmission doit s'arrêter
<Type>	Spécifie le format de transmission utilisé

15.2 Codage de type texte

15.2.1 Correspondance des mots clés

Voir le Tableau 94.

15.2.2 Codage des paramètres

Voir aussi 6.4.4 pour le codage des paramètres non dépendants du service utilisés.

15.2.2.1 Paramètre Service-id

Le paramètre Service-id est codé comme une chaîne STRING dont la valeur est constante et égale à "TLX".

Syntaxe:

<Service-id-parameter> := "TLX"

TABLEAU 94/T.611

Codage de type texte des éléments syntaxiques additionnels pour le service télex

(Le signe ↵ est le caractère "ligne suivante")

Élément syntaxique	Couple mot clé/paramètre
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvTlx>	"CVTLX" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵

15.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING dont la valeur est constante et égale à "STD".

Syntaxe:

<Type-id-parameter> := "STD"

NOTE – Le format de transmission format STD est défini dans la Recommandation UIT-T S.1.

15.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"ASCII" ASCII étendu APPLI/COM
 "ASCII437" ASCII étendu APPLI/COM
 "T.50" ASCII normalisé APPLI/COM

Syntaxe:

<Convert-id-parameter> := <Ascii> | "T.50"

<Ascii>:= "ASCII" | STRING ("ASCII" + <Code-page>)

<Code-page> := <digit> <digit> <digit>

<digit> := "0" | ... | "9"

15.2.2.4 Paramètres File-of-addrspec et Address

Le paramètre File-of-addrspec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Addrspec, qui contiennent le paramètre Address.

Le paramètre Address est codé comme une chaîne STRING formant le numéro d'appel. Lorsqu'on doit procéder à une vérification de l'identification de l'abonné, l'indicatif du terminal récepteur précédé du caractère égal ("=") doit être introduit après le numéro d'appel.

On peut aussi donner un nom de substitution à la place de l'identificateur de terminal, mais il faut alors que ce nom soit précédé du caractère "&". On suppose que l'application de communication sait décoder le nom de substitution spécifié.

NOTE – Si l'on veut procéder à une vérification de l'identification de l'abonné, l'indicatif complet doit être composé après le signe "=" et pas uniquement la composante alphabétique du récepteur.

Syntaxe:

```
<File-of-addrspec> :=          PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Addrsec-parameter>

<Addrsec-parameter> :=        <Address-parameter>

<Address-parameter> :=        <Telex-address> | ("&" <Alias>)

<Telex-address> :=            STRING
-- La chaîne doit représenter une adresse télex valide

<Alias> :=                     STRING
```

15.2.2.5 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre File. Le paramètre File lui-même est codé comme un trajet PATH pointant sur le fichier transféré.

Syntaxe:

```
<File-of-filespec> :=          PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Filespec-parameter>

<Filespec-parameter> :=        <File-parameter> "," <File-conv> ["," <File-type>]

<File-parameter> :=           PATH
-- Trajet vers le fichier transféré

<File-conv> :=                 <Convert-id-parameter>
-- Spécifie le format de transfert du fichier

<File-type> :=                 <Type-id-parameter>
-- Spécifie le format de transmission du fichier
```

15.2.2.6 Paramètre Sub-address

Le paramètre Sub-address est codé comme une chaîne NUMERIC-STRING qui représente une sous-adresse.

Syntaxe:

```
<Sub-address-parameter> :=    NUMERIC-STRING
```

15.3 Fonctionnalités additionnelles

15.3.1 Fonctions: Send et SendAck

Voir les Tableaux 95 et 96.

TABLEAU 95/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour le service télex

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi à un destinataire							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
Envoi à un ou à plusieurs destinataires							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Spécifie une liste de destinataires
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement, ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement, ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 96/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour le service télex par télétex

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Sous-adresse de l'expéditeur
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<From>	+	o	I	FROM	NUMERIC-STRING	Première page	Première page à envoyer réellement, ne s'applique qu'aux fichiers textuels
<To>	+	o	I	TO	NUMERIC-STRING	Dernière page	Dernière page à envoyer réellement, ne s'applique qu'aux fichiers textuels
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

15.3.2 Fonction: Receive

Voir le Tableau 97.

TABLEAU 97/T.611

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour le service télex par télétext

Elément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Spécifie le numéro de téléphone de l'expéditeur
<CvTlx>	B	o	I	CVTLX	<Convert-id-parameter>	"T.50"	Format de transfert souhaité par l'application locale
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Sous-adresse du destinataire; lorsqu'elle est spécifiée dans la demande, elle sert de sélecteur de recherche
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance par l'application locale. En cas de réception d'un seul fichier, le trajet doit être conservé par l'application de communication. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

15.4 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Lorsqu'une application de communication prend en charge le service télex, elle doit spécifier l'identificateur Type-id pris en charge (formats de transmission des documents) dans le descripteur CA-Descriptor (voir 9.5). Le Tableau 98 montre les valeurs que le descripteur CA peut prendre.

TABLEAU 98/T.611

Valeurs supplémentaires du descripteur CA-Descriptor pour le service télex

Mot clé	Paramètre	L'application de communication déclare que
TX	"STD"	le service de base est assuré

16 Service: Courrier électronique

La présente Recommandation traite également de l'envoi ou de la réception d'informations par courrier électronique. Il ne s'agit pas ici d'assurer la commande intégrale des services de courrier électronique à travers l'interface.

Les définitions ci-après permettent en outre l'accès au système de messagerie (MHS) décrit dans les Recommandations UIT-T de la série X.400. En termes de messagerie, l'interface assure l'accès à la messagerie interpersonnelle.

La mise en œuvre de l'application de communication peut reposer sur d'autres interfaces de prise en charge du courrier électronique comme indiqué à la Figure 15. L'application de communication représentée dans cette figure utilise une interface de courrier électronique sous-jacente du fournisseur du service de courrier électronique. En général, l'application de communication est libre d'utiliser une interface de courrier électronique *quelconque* pour accéder et commander les services de courrier électronique.

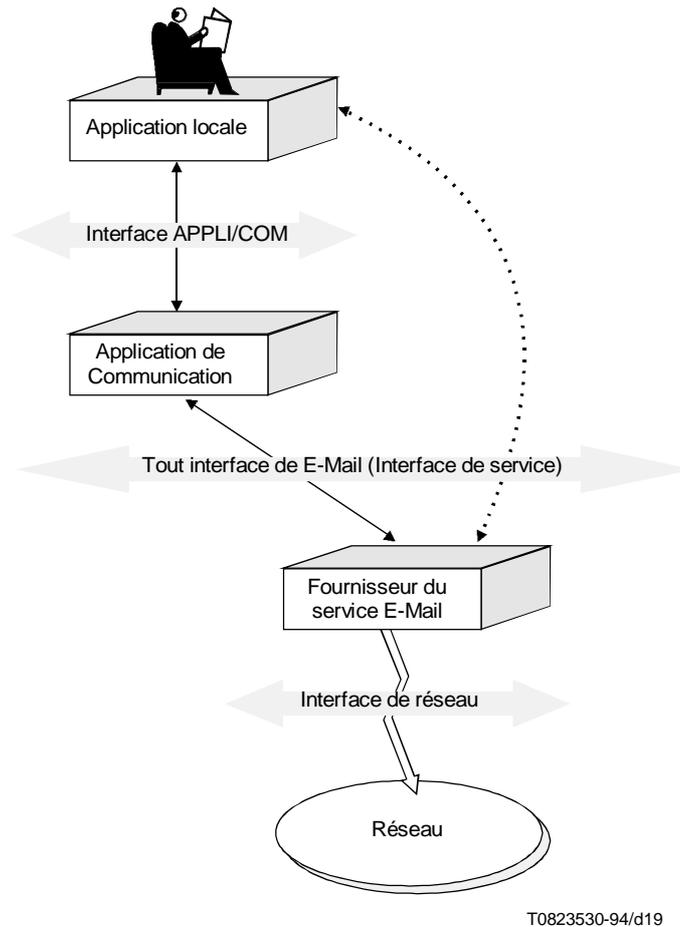


FIGURE 15/T.611

Modèle d'accès pour le courrier électronique

Les systèmes de messagerie et autres systèmes de courrier électronique normalisent la syntaxe et la sémantique des messages de personne à personne (IPM) (*interpersonal messages*) ainsi que la syntaxe et la sémantique des notifications de personne à personne (IPN) (*interpersonal notification*)⁹⁾ et des rapports de remise (DR) (*delivery report*).

⁹⁾ La Recommandation UIT-T X.420 normalise pour les systèmes de messagerie (MHS) la syntaxe des messages de personne à personne et des notifications de personne à personne. Dans la version de 1984 de la Recommandation UIT-T X.420, les notifications sont appelées SR-UAPDU.

Un destinataire peut créer et envoyer une notification de personne à personne automatiquement ou sur demande de l'application locale (notification explicite). Les notifications de personne à personne (IPN) sont de deux sortes:

- les notifications de réception (RN) (*receipt notification*): le message de personne à personne a été reçu;
- les notifications de non-réception (NRN) (*non receipt notification*): le message de personne à personne n'a pas été remis; dans ce cas, le message non remis peut être transmis avec la notification NRN.

A l'inverse de la notification de personne à personne, le rapport de remise est créé par le système de transfert de messages. On distingue:

- les rapports de remise (DR); et
- les rapports de non-remise (NDR) (*non-delivery report*).

Dans la description Send et SendAck TDD, l'application locale peut préciser si une notification de personne à personne ou si rapport de remise doit être produit pour chaque destinataire. Pour de plus amples détails voir 16.2.2.6, 16.4 et 16.5.

Les systèmes de messagerie et autres systèmes de courrier électronique permettent le transfert des messages de personne à personne complets – comportant des parties en-tête et corps – comme s'il s'agissait de la partie corps d'un message.

La présente Recommandation permet le transfert de ces parties corps dans les descriptions Send (SendAck) TDD et Receive TDD au moyen d'un paramètre Type-ID spécifique (le paramètre Type-id "MESSAGE", voir 16.2.2.2). Comme le contenu de cette partie corps est codé¹⁰, la présente Recommandation permet de décoder et de coder cette partie corps au moyen de descriptions TDD EncodeIPM et DecodeIPM.

Les descriptions TDD EncodeIPM et DecodeIPM doivent être mises en œuvre en utilisant l'élément <ExtendTDD>. Les extensions sont respectivement:

- <EncodeIPMExtension>; et
- <DecodeIPMExtension>.

Les descriptions TDD EncodeIPM et DecodeIPM sont décrites en 16.3.3.

16.1 Eléments syntaxiques propres au service

Les extensions dépendantes du service de la syntaxe générique des descriptions TDD du 6.1 sont indiquées ci-après:

```
<ServiceDependentKeywordsSend> :=
    <S-RecipientSpec> [<S-OriginatorSpec>]
    ((<Document> <Convert> [<Type>]) | <DocumentSpec>)
    <IpmId> [<Alternate>] [<ContType>] [<DiscloRec>] [<ExpiryTime>]
    [<ImplicitConv>] [<Importance>] [<Language>] [<Priority>]
    [<RelatedSpec>] [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>]
    [<UserInfo>]

<ServiceDependentKeywordsSendAck> :=
    <S-RecipientSpec> [<S-OriginatorSpec>]
    ((<Document> <Convert> [<Type>]) | <DocumentSpec>)
    <IpmId> [<Alternate>] [<ContType>] [<DiscloRec>] [<ExpiryTime>]
    [<ImplicitConv>] [<Importance>] [<Language>] [<Priority>]
    [<RelatedSpec>] [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>]
    [<UserInfo>] [<MsgSubId>] [<SubmitTime>]

<ServiceDependentKeywordsReceive> :=
    <R-RecipientSpec> [<R-OriginatorSpec>]
    ((<Document> <Convert> <Type>) | <DocumentSpec>)
    <IpmId> [<RelatedSpec>] [<ContType>] [<ExpiryTime>] [<Forwarded>]
    [<Importance>] [<IncompleteCopy>] [<Language>] [<Priority>]
    [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>] [<SubmitTime>]
    [<UserInfo>]
```

¹⁰ Dans le cas d'un système de messagerie (MHS), le message IPM est codé conformément au taux d'erreurs sur les bits de l'ASN.1.

```

<ExtendSubFunctionKeywords> :=
    <EncodeIPMExtension> | <DecodeIPMExtension>

<EncodeIPMExtension> :=
    <EncodeIPMSubFunction>
    ((<Document> <Convert> [<Type>]) | <DocumentSpec>) <Message>
    <S-RecipientSpec> [<S-OriginatorSpec>] <IpmId> [<ContType>] [<ExpiryTime>]
    [<Importance>] [<Priority>] [<RelatedSpec>] [<ReplyId>] [<ReplyTime>]
    <S-Originator> [<Sensitivity>] [<Subject>] [<UserInfo>]

<DecodeIPMExtension> :=
    <DecodeIPMSubFunction>
    <Message> ((<Document> <Convert> <Type>) | <DocumentSpec>)
    <S-RecipientSpec> [<S-OriginatorSpec>] <IpmId> [<ContType>] [<ExpiryTime>]
    [<Forwarded>] [<Importance>] [<Language>] [<Priority>] [<RelatedSpec>]
    [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>] [<UserInfo>]

```

Les éléments syntaxiques pour les services de courrier électronique sont décrits dans le Tableau 99. Pour une mise en correspondance des éléments de service du système de messagerie avec les éléments syntaxiques utilisés dans la présente Recommandation, se reporter au Tableau 112 du 16.6.

16.2 Codage de type texte

16.2.1 Correspondance des mots clés

Voir le Tableau 100.

16.2.2 Codage des paramètres

Voir aussi 6.4.4 pour le codage des paramètres non dépendants du service utilisés.

16.2.2.1 Paramètre Service-id

Le paramètre Service-id est codé comme une chaîne STRING dont la valeur est constante et égale à "EMAIL".

Syntaxe:

```
<Service-id-parameter> := "EMAIL"
```

16.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"STD"	Partie de corps de texte IA5
"TELETEX"	Partie corps télétext
"G3FAX"	Partie corps télécopie du groupe 3
"G4CLASS1"	Partie corps télécopie du groupe 4
"VIDEOTEX"	Partie corps vidéotex
"MESSAGE"	Partie corps contenant un message de personne à personne (composé d'un en-tête et de parties corps) telle que transférée par le système de courrier électronique
"BILATERAL"	Contenu de partie corps défini de manière bilatérale
"NATIONAL"	Contenu de partie corps défini au plan national
"ODA"	Partie corps ODA

Syntaxe:

```
<Type-id-parameter> := "STD" | "TELETEX" | "G3FAX" | "G4CLASS1" | "VIDEOTEX" | "MESSAGE" |
    "BILATERAL" | "NATIONAL" | "ODA"11)
```

¹¹⁾ La Recommandation UIT-T X.420 définit pour les systèmes de messagerie (MHS) un jeu de paramètres se rapportant à chaque partie corps. Ces paramètres sont transférés à l'aide du paramètre Filespec (voir 16.2.2.8).

TABLEAU 99/T.611

Eléments syntaxiques supplémentaires pour les services de courrier électronique

Elément syntaxique	Objet
<Alternate>	Spécifie si d'autres destinataires sont autorisés à recevoir le message
<ContType>	Spécifie le type de contenu du message échangé
<Convert>	Spécifie le format de transfert à utiliser
<DecodeIPMSubFunction>	Identifie la sous-fonction DecodeIPM
<DiscloRec>	Divulgaration des destinataires
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception (ou à échanger avec les sous-fonctions EncodeIPM et DecodeIPM)
<EncodeIPMSubFunction>	Identifie la sous-fonction EncodeIPM
<ExpiryTime>	Date/heure d'expiration du message
<Forwarded>	Indication retransmise automatiquement
<ImplicitConv>	Conversion implicite
<Importance>	Importance du contenu du message
<IncompleteCopy>	Indique que la copie du système de messagerie est incomplète
<IpmId>	ID du message
<Language>	Identification de la langue utilisée
<Message>	Trajet du message codé conformément au taux d'erreurs sur les bits de l'ASN.1 des sous-fonctions EncodeIPM et DecodeIPM
<MsgSubId>	Identificateur de soumission d'un message, renvoyé par le fournisseur du service de courrier électronique après qu'il a accepté la demande
<Priority>	Priorité du contenu du message
<R-OriginatorSpec>, <S-OriginatorSpec>	Trajet vers le fichier contenant les spécifications d'adresse de l'expéditeur, de(s) visa(s) d'expédition et réponse à ou aux utilisateurs du message
<R-RecipientSpec>, <S-RecipientSpec>	Trajet vers le fichier contenant les spécifications d'adresse de tous les destinataires désignés du message, à savoir des destinataires principaux, des destinataires de copies et des destinataires de copies muettes
<RelatedSpec>	Trajet vers le fichier contenant les spécifications associées Relatedspecs de tous les messages liés au message actuel
<ReplyId>	ID du message auquel répond le présent message
<ReplyTime>	Date-heure à laquelle le(s) destinataire(s) devrai(en)t avoir répondu aux utilisateurs qui donnent les visas d'expédition
<Sensitivity>	Confidentialité du contenu du message
<Subject>	Objet du message
<SubmitTime>	Date-heure à laquelle le fournisseur du service de courrier électronique a traité le message (date-heure de soumission du message)
<Type>	Spécifie le format de transmission utilisé
<UserInfo>	Identificateur facultatif d'enveloppe fourni par l'utilisateur, qui est transmis parallèlement au message. Peut être utilisé pour les besoins d'identification propres à l'utilisateur

TABLEAU 100/T.611

Codage de type texte des éléments syntaxiques additionnels pour les services de courrier électronique

(Le signe ↵ est le caractère "ligne suivante")

Élément syntaxique	Couple mot clé/paramètre
<Alternate>	"ALTERNATE" ":" <Boolean-parameter> ↵
<ContType>	"CONT-TYPE" ":" <Content-type-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<DecodeIPMSubFunction>	"SUBFUNC" ":" "DecodeIPM" ↵
<DiscloRec>	"DISCLO-REC" ":" <Boolean-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<EncodeIPMSubFunction>	"SUBFUNC" ":" "EncodeIPM" ↵
<ExpiryTime>	"EXPIRYTIME" ":" <Date-time-parameter> ↵
<Forwarded>	"FORWARDED" ":" <Boolean-parameter> ↵
<ImplicitConv>	"IMPLICIT-CONV" ":" <Boolean-parameter> ↵
<Importance>	"IMPORTANCE" ":" <Importance-parameter> ↵
<IncompleteCopy>	"INC-COPY" ":" <Boolean-parameter> ↵
<IpmId>	"IPM-ID" ":" <Ipm-id-parameter> ↵
<Language>	"LANGUAGE" ":" <Language-id-parameter> ↵
<Message>	"MESSAGE" ":" <Path-parameter> ↵
<MsgSubId>	"MSG-SUB-ID" ":" <Msg-sub-id-parameter> ↵
<Priority>	"PRIORITY" ":" <Priority-parameter> ↵
<R-OriginatorSpec>	"ADDRESS" ":" "@" <File-of-originatorspec> ↵
<R-RecipientSpec>	"RECIPIENT" ":" "@" <File-of-r-recipientspec> ↵
<RelatedSpec>	"RELATED" ":" "@" <File-of-relatedspec> ↵
<ReplyId>	"REPLYID" ":" <Ipm-id-parameter> ↵
<ReplyTime>	"REPLYTIME" ":" <Date-time-parameter> ↵
<S-OriginatorSpec>	"ORIGINATOR" ":" "@" <File-of-originatorspec> ↵
<S-RecipientSpec>	"ADDRESS" ":" "@" <File-of-s-recipientspec> ↵
<Sensitivity>	"SENSITIVITY" ":" <Sensitivity-parameter> ↵
<Subject>	"SUBJECT" ":" <Subject-parameter> ↵
<SubmitTime>	"SUBMITTIME" ":" <Date-time-parameter> ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UserInfo>	"USERINFO" ":" <Userinfo-parameter> ↵

16.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"ASCII"	ASCII étendu APPLI/COM
"ASCII437"	ASCII étendu APPLI/COM
"T.50"	ASCII normalisé APPLI/COM
"T.61"	Format télétext APPLI/COM
"TIFF"	Format TIFF APPLI/COM
"VOID"	Pas de conversion à effectuer
"PROBE"	Spécifie un document vide (c'est-à-dire aucun document) pour les procédures de test de remise

L'utilisation du paramètre Convert-id dépend aussi du paramètre Type-id choisi (voir le Tableau 101).

TABLEAU 101/T.611

Valeurs autorisées du paramètre Convert-id dépendantes du paramètre Type-id

Type-id	Valeurs du paramètre Convert-id pour le trafic sortant	Valeurs du paramètre Convert-id pour le trafic entrant
STD	T.50	
TELETEX	ASCII, ASCII _{xxx} ^{a)} , T.50, T.61	
G3FAX, G4CLASS1	ASCII, ASCII _{xxx} , T.50, TIFF	TIFF
VIDEOTEX, MESSAGE, BILATERAL, NATIONAL, ODA	VOID	
a) xxx est un code-page, déclaré dans l'ICE; par exemple, ASCII437 si le code-page 437 a été déclaré.		

Syntaxe:

<Convert-id-parameter> :=	<Convert-probe> <Convert-std> <Convert-ttx> <Convert-img> <Convert-bin>
<Convert-probe> :=	"PROBE" -- Mode spécial, dans lequel aucun document n'est transmis; peut seulement être envoyé
<Convert-std> :=	"T.50"
<Convert-ttx> :=	<Ascii> "T.50" "T.61"
<Convert-img> :=	<Ascii> "T.50" "TIFF"
<Convert-bin> :=	"VOID"
<Ascii> :=	"ASCII" STRING ("ASCII" + <Code-page>)
<Code-page> :=	<digit> <digit> <digit>
<digit> :=	"0" ... "9"

16.2.2.4 Adresses MHS

Les adresses MHS sont désignées par le paramètre Originatorspec, les paramètres R-Recipientspec et S-Recipientspec ou par le paramètre Ipm-id (voir 16.2.2.5, 16.2.2.6 et 16.2.2.11).

La syntaxe des adresses MHS, à savoir Mhs-or-descriptor, Mhs-or-name, Mhs-dl-name et Mhs-or-address, est spécifiée ci-après.

L'adresse MHS-or-address est codée comme une chaîne STRING représentant une compilation d'éléments d'adresse, chacun de ces éléments spécifiant une partie de l'adresse.

Un élément d'adresse est composé d'un identificateur de champ d'adresse suivi d'un séparateur (signe "=") et de l'attribut de champ d'adresse.

Le Tableau 102 donne la liste des identificateurs de champ d'adresse prédéfinis pour les services de messagerie dans le cadre de la présente Recommandation¹²⁾.

NOTE – Le plan d'adressage du Tableau 102 peut aussi être utilisé pour d'autres environnements de courrier électronique s'il y a lieu. La création d'autres éléments d'adresse prédéfinis est possible au moyen de la description syntaxique au format style-BNF de l'élément <Private-addr-element> ci-dessous.

TABLEAU 102/T.611

Identificateurs et attributs de champ d'adresse

Identificateur de champ d'adresse	Attribut de champ d'adresse	Longueur maximale du champ d'attribut
C	CountryName	3
A	AdministrationDomainName	16
P	PrivateDomainName	16
O	OrganizationName	64
OUn (0 ≤ n ≤ 4)	OrganizationUnitName	32
Q	GenerationQualifier	3
S	Surname	40
G	GivenName	16
I	Initials	5
X121	X.121Address	16
T-ID	TerminalIdentifier	24
PD-BOX	PostalOfficeBoxAddress	60

Syntaxe:

-- Syntaxe de <Mhs-or-address> (à laquelle renvoient d'autres syntaxes de paramètres)

<Mhs-or-address> :=

```
"/" <Mandatory-address-element> { "/" <Mandatory-address-element> }
{ "/" <Optional-address-element> } { "/" <Private-addr-element> }
-- à utiliser avec les autres systèmes de messagerie
-- un élément <Mandatory-address-element> au moins doit être présent
```

<Mandatory-address-element> :=

```
("C" "=" STRING (SIZE(1..3))) | -- Nom de pays
("A" "=" STRING (SIZE(1..16))) | -- Nom de domaine administratif
("P" "=" STRING (SIZE(1..16))) | -- Nom de domaine privé
("O" "=" STRING (SIZE(1..64))) | -- Nom d'organisation
-- voir également le Tableau 102 pour les affectations
```

<Optional-address-element> :=

```
("OU"<n> "=" STRING (SIZE(1..32))) | -- Nom d'unité d'organisation
("Q" "=" STRING (SIZE(1..3))) | -- Qualificatif de génération
("S" "=" STRING (SIZE(1..40))) | -- Nom patronymique
```

¹²⁾ Chaque identificateur de champ d'adresse désigne un attribut de nom d'O/R-Name spécifié dans la Recommandation UIT-T X.402.

```

("G" "=" STRING (SIZE(1..16))) |          -- Prénom
("I" "=" STRING (SIZE(1..5))) |          -- Initiales
("X121" "=" STRING (SIZE(1..16))) |      -- Adresse X.121
("T-ID" "=" STRING (SIZE(1..24))) |      -- Identificateur de terminal
("PD-BOX" "=" STRING (SIZE(1..604))) |   -- Adresse de la boîte postale
-- voir également le Tableau 102 pour les affectations

<n> :=                                     "0" | ... | "4"

<Private-addr-element> :=                 <Field-identifiant> "=" <Field-attribute>
-- Élément d'adresse défini de manière privée ou additionnelle

-- Syntaxe de <Mhs-or-descriptor> (à laquelle renvoient d'autres syntaxes de paramètre)

<Mhs-or-descriptor> :=                   (<Mhs-or-name> ["+" [<Mhs-free-name>] ["+" <Mhs-phone-number>]]) |
("+" <Mhs-free-name> ["+" <Mhs-phone-number>])

<Mhs-free-name> :=                       STRING (SIZE(1..64))

<Mhs-phone-number> :=                   STRING (SIZE(1..32))

-- Syntaxe de <Mhs-or-name> (à laquelle renvoient d'autres syntaxes de paramètres)

<Mhs-or-name> :=                         (<Mhs-or-address> ["+" <Mhs-dl-name>]) | ("+" <Mhs-dl-name>)

<Mhs-dl-name> :=                         STRING (SIZE(1..64))
-- Désigne le nom de la liste de distribution conforme à la
-- Recommandation UIT-T X.402

```

NOTE – L'élément <Mhs-dl-name> est différent de l'élément <Alias> utilisé dans certaines syntaxes, car le nom de substitution est développé localement (dans l'application de communication) alors que l'élément est développé dans le système de transfert de messages (MTS).

16.2.2.5 Construction File-of-originatorspec

La construction File-of-originatorspec spécifie l'expéditeur du message. Elle est codée comme un trajet PATH. Le trajet donné doit renvoyer à un fichier contenant un ou plusieurs paramètres Originatorspec.

Syntaxe:

```

<File-of-originatorspec> :=              PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Originatorspec-parameter>

<Originatorspec-parameter> :=           <Addr-descriptor> | <Mhs-or-descriptor> "," <O-type>
-- pour la définition de <Mhs-or-descriptor> voir 16.2.2.4

<Addr-descriptor> :=                   STRING
-- Contenu demandé par le système de courrier électronique sous-jacent
-- (non-système de messagerie);
-- ne doit pas contenir de caractères ", "

<O-type> :=                             "Authorizing" | "Originator" | "ReplyTo"

```

16.2.2.6 File-of-r-recipientspec et File-of-s-recipientspec

Les constructions File-of-r-recipientspec et File-of-s-recipientspec spécifient les destinataires du message. Elles sont codées comme un trajet PATH. Le trajet donné doit renvoyer à un fichier contenant un ou plusieurs paramètres R-recipientspec ou S-recipientspec respectivement.

Syntaxe:

```

<File-of-r-recipientspec> :=            PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <R-recipientspec-parameter>

```

<File-of-s-recipientspec> := PATH
 -- *Le trajet pointe sur un fichier contenant un ou plusieurs paramètres*
 -- <S-recipientspec-parameter>

<R-recipientspec-parameter> :=
 <Addr-descriptor> | <Mhs-or-descriptor>
 "," <R-type> ["," [<Reply>] ["," [<Notify>] ["," <Report>]]]
 -- *côté réception*
 -- *pour la définition de <Mhs-or-descriptor> voir 16.2.2.4*

<S-recipientspec-parameter> :=
 ("&" <Alias>) | <Addr-descriptor> | <Mhs-or-descriptor>
 "," <S-type> ["," [<Reply>] ["," [<Notify>] ["," <Report>]]]
 -- *Le côté émission peut spécifier un élément de substitution (local) à*
 -- *la place d'un descripteur*
 -- *pour la définition de <Mhs-or-descriptor> voir 16.2.2.4*

<Alias> := STRING
 -- *L'élément de substitution ne doit pas contenir de caractères "," ou ";"*

<Addr-descriptor> := STRING
 -- *Contenu demandé par le système de courrier électronique sous-jacent*
 -- *(non-système de messagerie);*
 -- *ne doit pas contenir de caractères ","*

<R-type> := "Primary" | "Copy" | "Blind" | "Intended"

<S-type> := "Primary" | "Copy" | "Blind"

<Reply> := "NoReply" | "Reply"
 -- *NoReply (non-réponse) est la valeur par défaut lorsque non spécifiée*

<Notify> := "NoNotify" | "NotifyNotReceived" | "NotifyReceived" | "NotifyExplicit"
 -- *NoNotify (non-notification) est la valeur par défaut lorsque non spécifiée*

<Report> := "NoReport" | "BasicReport" | "ConfirmedReport"¹³⁾
 -- *NoReport (pas de rapport) est la valeur par défaut lorsque non spécifiée*

16.2.2.7 Construction File-of-relatedspec

La construction File-of-relatedspec spécifie les identificateurs d'un message de personne à personne (IPM-ID), auxquels un message se rapporte. Elle est codée comme un trajet PATH. Ce trajet doit renvoyer à un fichier contenant un ou plusieurs paramètres Relatedspec.

Syntaxe:

<File-of-relatedspec> := PATH
 -- *Le trajet pointe sur un fichier contenant un ou plusieurs paramètres*
 -- <Relatedspec-parameter>

<Relatedspec-parameter> := <Ipm-Id-parameter> "," <Relation>
 -- *pour la définition du paramètre Ipm-id voir 16.2.2.11*

<Relation> := "Reference" | "Obsolete"

16.2.2.8 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre File. Le paramètre File est lui-même codé comme un trajet PATH pointant sur le fichier transféré.

Lorsque l'élément spécial "PROBE" du paramètre Convert-id est spécifié, le paramètre File peut être omis (voir aussi 16.2.2.3).

¹³⁾ Dans le cas où MHS "BasicReport" correspond à un rapport de non-remise et "ConfirmedReport" correspond au rapport de la Recommandation UIT-T X.411.

Syntaxe:

```
<File-of-filespec> :=          PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Filespec-parameter>

<Filespec-parameter> :=      <File-parameter> ", " <File-conv>
[[", " <File-type>] [", " <Body-part-parameter>]]

<File-parameter> :=          PATH
-- Trajet vers le fichier transféré

<File-conv> :=                <Convert-id-parameter>
-- Spécifie le format de transfert du fichier

<File-type> :=                <Type-id-parameter>
-- Spécifie le format de transmission du fichier

<Body-part-parameter> :=      STRING
-- Syntaxe non déterminée dans l'attente d'un complément d'étude
```

16.2.2.9 Paramètre Content-type

Le paramètre Content-type est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"mhsIPM84"	Message de personne à personne (IPM) conforme au protocole P2 défini dans la Recommandation UIT-T X.420 (1984)
"mhsIPM88"	Message de personne à personne (IPM) conforme au protocole P2 défini dans la Recommandation UIT-T X.420 (1988)
"mhsIPN84"	Notification de personne à personne (IPN) conforme au protocole P2 défini dans la Recommandation UIT-T X.420 (1984)
"mhsIPN88"	Notification de personne à personne (IPN) conforme au protocole P2 défini dans la Recommandation UIT-T X.420 (1988)
"mhsDR"	Rapport de remise (DR) défini dans la Recommandation UIT-T X.420
"cmcIPM"	Réservé aux messages de personne à personne (IPM) conformes à l'interface commune d'appel de messagerie du courrier électronique
"cmcIPN"	Réservé aux notifications de personne à personne (IPN) conformes à l'interface commune d'appel de messagerie du courrier électronique
"cmcDR"	Réservé aux rapports de remise (DR) conformes à l'interface commune d'appel de messagerie du courrier électronique
autre	Réservé aux études futures

Syntaxe:

```
<Content-type-parameter> :=    "mhsIPM84" | "mhsIPM88" | "mhsIPN84" | "mhsIPN88" | "mhsDR"
                                | "cmcIPM" | "cmcIPN" | "cmcDR"
```

16.2.2.10 Paramètre Importance

Le paramètre Importance est codé comme une chaîne numérique NUMERIC-STRING prenant l'une des valeurs suivantes:

"0"	faible
"1"	normale (par défaut)
"2"	élevée

Syntaxe:

```
<Importance-parameter> :=      "0" | ... | "2"
```

16.2.2.11 Paramètre Ipm-id

Le paramètre Ipm-id est codé comme une chaîne STRING.

Dans les systèmes de messagerie, il a deux composantes:

- un identificateur unique, relatif à l'utilisateur (identificateur local);
- une adresse expéditeur/destinataire facultative qui identifie l'utilisateur.

Syntaxe:

<Ipm-id-parameter> := <Free-ipm-id> | <Mhs-ipm-id>

-- pour la définition de l'élément <Mhs-or-descriptor> voir 16.2.2.4

<Free-ipm-id> :=
STRING
-- Contenu demandé par le système de courrier électronique sous-jacent
-- (non-système de messagerie);
-- string ne doit pas contenir de caractères ", " ou " ; "

<Mhs-ipm-id> :=
<Local-ipm-id> [", " <Mhs-or-address>]
-- à utiliser dans les systèmes de messagerie
-- pour la définition de l'élément <Mhs-or-address> voir 16.2.2.4

<Local-ipm-id> :=
STRING (SIZE(1..64))
-- string ne doit pas contenir de caractères ", " ou " ; "

16.2.2.12 Paramètre Language-id

Le paramètre Language-id est codé comme une chaîne STRING comprenant 2 à 5 caractères. Pour l'attribution des codes de langue pour les systèmes de messagerie voir ISO 639.2.

Syntaxe:

<Language-id-parameter> := STRING (SIZE(2..5))

16.2.2.13 Paramètre Msg-sub-id

Le paramètre Msg-sub-id est codé comme une chaîne STRING. Le paramètre Msg-sub-id identifie de manière univoque et non ambiguë la soumission du message. Il est produit par le système de courrier électronique¹⁴.

Syntaxe:

<Msg-sub-id-parameter> := <Free-sub-id> | <Mhs-sub-id>

<Free-sub-id> :=
STRING
-- A utiliser dans les systèmes qui ne sont pas des systèmes de
-- messagerie

<Mhs-sub-id> :=
<Global-id> ", " <Local-id>
-- A utiliser dans les systèmes de messagerie

<Global-id> :=
"/" "C" "=" STRING (SIZE(1..3)) -- Nom de pays
"/" "A" "=" STRING (SIZE(1..16)) -- Nom de domaine
-- d'administration
["/" "P" "=" STRING (SIZE(1..16))] -- Nom de domaine privé
-- Voir aussi la définition de l'élément <Mhs-or-address> au 16.2.2.4

<Local-id> :=
STRING (SIZE(1..32))

16.2.2.14 Paramètre Priority

Le paramètre Priority est codé comme une chaîne numérique NUMERIC-STRING prenant l'une des valeurs suivantes:

"0"	normal (par défaut)
"1"	non urgent
"2"	urgent

¹⁴ Dans le cas d'un système de messagerie (MHS), le paramètre Msg-sub-id correspond à l'identificateur MPDUIdentifier (Recommandation UIT-T X.411 de 1984), renommé MTSIdentifier dans la version de 1988 de la Recommandation UIT-T X.411.

Syntaxe:

<Priority-parameter> := "0" | "1" | "2"

16.2.2.15 Paramètre Subject

Le paramètre Subject est codé comme une chaîne STRING.

Syntaxe:

<Subject-parameter> := STRING (SIZE(1..128))

16.2.2.16 Paramètre Sensitivity

Le paramètre Sensitivity est codé comme une chaîne numérique NUMERIC-STRING prenant l'une des valeurs suivantes:

"0"	pas de confidentialité spécifiée (par défaut)
"1"	personnel
"2"	privé
"3"	confidentiel entreprise

Syntaxe:

<Sensitivity-parameter> := "0" | ... | "3"

16.2.2.17 Paramètre Userinfo

Le paramètre Userinfo est codé comme une chaîne STRING.

Syntaxe:

<Userinfo-parameter> := STRING (SIZE(1..16))

16.3 Messagerie de personne à personne

16.3.1 Fonctions: Send et SendAck

La description TDD de réponse de la fonction SendAck doit être produite par l'application de communication dès que le document (le message) a été soumis au fournisseur du service de courrier électronique de l'expéditeur et que le fournisseur du service de courrier électronique a renvoyé l'élément <MsgSubId> à l'application de communication. En d'autres termes: la soumission fait l'objet d'un accusé de réception, et non pas la transmission du message. Voir les Tableaux 103 et 104.

16.3.2 Fonction: Receive

Voir le Tableau 105.

16.3.3 Fonctions: EncodeIPM et DecodeIPM

Les fonctions TDD EncodeIPM et DecodeIPM permettent à une application locale de coder ou de décoder un message de personne à personne IPM – contenant des parties en-tête et corps – en une/à partir d'une partie corps de type "MESSAGE" (voir 16.2.2.2).

La procédure de codage/décodage nécessite la fourniture par l'application locale d'une description TDD contenant des éléments syntaxiques renvoyant à l'en-tête du message de personne à personne IPM à coder/décoder ainsi que des éléments syntaxiques déclenchant l'opération de codage/décodage. Le Tableau 106 et le Tableau 107 montrent respectivement les extensions EncodeIPM TDD et les extensions DecodeIPM TDD apportées à l'entité <ExtendTDD>.

TABLEAU 103/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour l'accès aux services de courrier électronique

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<IpmId>	B	m	I	IPM-ID	<Ipm-id-parameter>	–	Identificateur de message
<S-RecipientSpec>	B	m	I	ADDRESS	"@" <File-of-s-recipient-spec>	–	Trajet du fichier contenant les spécifications Recipientspecs de tous les destinataires supposés
<S-OriginatorSpec>	B	m	I	ORIGINATOR	"@" <File-of-originator-spec>	–	Nom de trajet du fichier contenant les coordonnées de l'expéditeur et des utilisateurs qui donnent les visas d'expédition
<Alternate>	B	o	I	ALTERNATE	<Boolean-parameter>	"Non"	Autre(s) destinataire(s) autorisé(s)
<ContType>	B	o	I	CONT-TYPE	<Content-type-parameter>	"mhsIPM84"	Type de contenu
<DiscloRec>	B	o	I	DISCLO-REC	<Boolean-parameter>	"Non"	Divulgarion du ou des destinataires
<ExpiryTime>	B	o	I	EXPIRYTIME	<Date-time-parameter>	–	Expiration du message (date/heure d'expiration)
<ImplicitConv>	B	o	I	IMPLICIT-CONV	<Boolean-parameter>	"Oui"	Conversion implicite
<Importance>	B	o	I	IMPORTANCE	<Importance-parameter>	"1"	Importance du contenu
<Language>	B	o	I	LANGUAGE	<Language-id-parameter>	–	Identificateur de la langue utilisée
<Priority>	B	o	I	PRIORITY	<Priority-parameter>	"0"	Priorité du contenu

TABLEAU 103/T.611 (fin)

Fonctionnalités additionnelles de la fonction <SendTDD> pour l'accès aux services de courrier électronique

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-Relatedspec>	–	Nom de trajet du fichier contenant les spécifications Relatedspecs de tous les messages liés au présent message
<ReplyId>	B	o	I	REPLYID	<Ipm-id-parameter>	–	Identification du message, pour lequel le présent message est une réponse
<ReplyTime>	B	o	I	REPLYTIME	<Date-time-parameter>	–	Date-heure à laquelle le(s) destinataire(s) doit (doivent) répondre aux utilisateurs qui donnent les visas d'expédition
<Sensitivity>	B	o	I	SENSITIVITY	<Sensitivity-parameter>	"0"	Confidentialité du contenu
<Subject>	B	o	I	SUBJECT	<Subject-parameter>	–	
<Userinfo>	+	o	I	USERINFO	<Userinfo-parameter>	–	Identificateur facultatif d'enveloppe fourni par l'utilisateur
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 104/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour l'accès aux services de courrier électronique

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<IpmId>	B	m	I	IPM-ID	<Ipm-id-parameter>	–	Identificateur de message
<S-RecipientSpec>	B	m	I	ADDRESS	"@" <File-of-s-recipient-spec>	–	Trajet du fichier contenant les spécifications Recipientspecs de tous les destinataires supposés
<S-OriginatorSpec>	B	m	I	ORIGINATOR	"@" <File-of-originator-spec>	–	Nom de trajet du fichier contenant les coordonnées de l'expéditeur et des utilisateurs qui donnent les visas d'expédition
<Alternate>	B	o	I	ALTERNATE	<Boolean-parameter>	"No"	Autre(s) destinataire(s) autorisé(s)
<ContType>	B	o	I	CONT-TYPE	<Content-type-parameter>	"mhsIPM84"	Type de contenu
<DiscloRec>	B	o	I	DISCLO-REC	<Boolean-parameter>	"No"	Divulgateur du ou des destinataires
<ExpiryTime>	B	o	I	EXPIRYTIME	<Date-time-parameter>	–	Expiration du message (date/heure d'expiration)
<ImplicitConv>	B	o	I	IMPLICIT-CONV	<Boolean-parameter>	"Yes"	Conversion implicite
<Importance>	B	o	I	IMPORTANCE	<Importance-parameter>	"1"	Importance du contenu
<Language>	B	o	I	LANGUAGE	<Language-id-parameter>	–	Identificateur de la langue utilisée
<MsgSubId>	B	o	O	MSG-SUB-ID	<Msg-sub-id-parameter>	–	Identificateur de soumission de message. Renvoyé par le fournisseur du service de courrier électronique après acceptation de la demande
<Priority>	B	o	I	PRIORITY	<Priority-parameter>	"0"	Priorité du contenu
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-Related-spec>	–	Nom de trajet du fichier contenant les spécifications Relatedspecs de tous les messages liés au présent message
ReplyId>	B	o	I	REPLYID	<Ipm-id-parameter>	–	ID du message auquel le présent message répond

TABLEAU 104/T.611 (fin)

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour l'accès aux services de courrier électronique

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<ReplyTime>	B	o	I	REPLYTIME	<Date-time-parameter>	–	Date-heure à laquelle le(s) destinataire(s) doit (doivent) répondre aux utilisateurs qui donnent les visas d'expédition
<Sensitivity>	B	o	I	SENSITIVITY	<Sensitivity-parameter>	"0"	Confidentialité du contenu
<Subject>	B	o	I	SUBJECT	<Subject-parameter>	–	
<SubmitTime>	B	o	O	SUBMITTIME	<Date-time-parameter>	–	Date-heure à laquelle la demande est traitée par le fournisseur du service de courrier électronique
<Userinfo>	+	o	I	USERINFO	<Userinfo-parameter>	–	Identificateur facultatif d'enveloppe fourni par l'utilisateur
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique à transmettre, remis dans le format spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 105/T.611

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour l'accès aux services de courrier électronique

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<IpmId>	B	m	O	IPM-ID	<Ipm-id-parameter>	–	Identificateur de message
<R-RecipientSpec>	B	m	I	RECIPIENT	"@" <File-of-r-recipient-spec>	–	Trajet du fichier contenant les spécifications Recipientspecs de tous les destinataires supposés
<R-OriginatorSpec>	B	m	I	ADDRESS	"@" <File-of-originator-spec>	–	Nom de trajet du fichier contenant les coordonnées de l'expéditeur et des utilisateurs qui donnent les visas d'expédition
<ContType>	B	o	O	CONT-TYPE	<Content-type-parameter>	–	Type de contenu
<ExpiryTime>	B	o	O	EXPIRYTIME	<Date-time-parameter>	–	Expiration du message (date-heure d'expiration)
<Forwarded>	B	o	O	FORWARDED	<Boolean-parameter>	–	Indication retransmise automatiquement
<Importance>	B	o	O	IMPORTANCE	<Importance-parameter>	–	Importance du contenu
<Language>	B	o	O	LANGUAGE	<Language-id-parameter>	–	Identificateur de la langue utilisée
<Priority>	B	o	O	PRIORITY	<Priority-parameter>	–	Priorité du contenu
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-related-spec>	–	Nom de trajet du fichier contenant les spécifications Relatedspecs de tous les messages liés au présent message
<ReplyId>	B	o	O	REPLYID	<Ipm-id-parameter>	–	ID du message auquel le présent message répond
<ReplyTime>	B	o	O	REPLYTIME	<Date-time-parameter>	–	Date-heure à laquelle le(s) destinataire(s) doit (doivent) répondre aux utilisateurs qui donnent les visas d'expédition
<Sensitivity>	B	o	O	SENSITIVITY	<Sensitivity-parameter>	"0"	Confidentialité du contenu
<Subject>	B	o	O	SUBJECT	<Subject-parameter>	–	

TABLEAU 105/T.611 (fin)

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour l'accès aux services de courrier électronique

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<SubmitTime>	B	o	O	SUBMITTIME	<Date-time-parameter>	–	Date-heure à laquelle la demande est traitée par le fournisseur du service de courrier électronique
<IncCopy>	+	o	O	INC-COPY	<Boolean-parameter>	–	Indicateur de copie incomplète
<Userinfo>	+	o	O	USERINFO	<Userinfo-parameter>	–	Identificateur d'enveloppe facultatif fourni par l'utilisateur
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance sur demande par l'application locale. En cas de réception d'un seul fichier, le nom doit être conservé par l'application de communication. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 106/T.611

**Éléments syntaxiques supplémentaires pour l'extension <EncodeIPMExtension>
des services de courrier électronique**

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
Éléments syntaxiques appartenant à la procédure de codage elle-même (description des tâches)							
<EncodeIPM-SubFunction>	B	m	I	SUBFUNC	"EncodeIPM"	--	
<Message>	B	m	I	MESSAGE	<Path-parameter>	--	Trajet vers le fichier qui va recevoir le message de personne à personne codé
<Minor>	+	o	O	MINOR	<Error-parameter>	-	
<Warning>	+	o	O	WARNING	<Error-parameter>	-	
Une seule partie corps doit être codée							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Format de transfert de la partie corps à coder
<Document>	B	m	I	FILENAME	<File-parameter>	-	Partie corps unique à coder, remise dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à coder
Une ou plusieurs parties corps doivent être codées							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Spécifie une liste de fichiers; utilise une syntaxe particulière
Éléments syntaxiques détenant les informations d'en-tête pour le message de personne à personne à coder							
<IpmId>	B	m	I	IPM-ID	<Ipm-id-parameter>	-	Identificateur de message
<S-RecipientSpec>	B	m	I	ADDRESS	"@" <File-of-s-recipient-spec>	-	Trajet du fichier contenant les spécifications Recipientspecs de tous les destinataires supposés
<S-OriginatorSpec>	B	m	I	ORIGINATOR	"@" <File-of-originator-spec>	-	Nom de trajet du fichier contenant les coordonnées de l'expéditeur et des utilisateurs qui donnent les visas d'expédition
<ContType>	B	o	I	CONT-TYPE	<Content-type-parameter>	"mhsIPM84"	Type de contenu
<ExpiryTime>	B	o	I	EXPIRYTIME	<Date-time-parameter>	-	Expiration du message (date-heure d'expiration)

TABLEAU 106/T.611 (fin)

**Eléments syntaxiques supplémentaires pour l'extension <EncodeIPMExtension>
des services de courrier électronique**

Elément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Importance>	B	o	I	IMPORTANCE	<Importance-parameter>	"1"	Importance du contenu
<Priority>	B	o	I	PRIORITY	<Priority-parameter>	"0"	Priorité du contenu
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-Relatedspec>	–	Nom de trajet du fichier contenant les spécifications Relatedspecs de tous les messages liés au présent message
<ReplyId>	B	o	I	REPLYID	<Ipm-id-parameter>	–	ID du message auquel le présent message répond
<ReplyTime>	B	o	I	REPLYTIME	<Date-time-parameter>	–	Date-heure à laquelle le(s) destinataire(s) doit (doivent) répondre aux utilisateurs qui donnent les visas d'expédition
<Sensitivity>	B	o	I	SENSITIVITY	<Sensitivity-parameter>	"0"	Confidentialité du contenu
<Subject>	B	o	I	SUBJECT	<Subject-parameter>	–	
<Userinfo>	+	o	I	USERINFO	<Userinfo-parameter>	–	Fourni par l'utilisateur

TABLEAU 107/T.611

**Éléments syntaxiques supplémentaires associés à l'extension <DecodeIPMExtension>
pour les services de courrier électronique**

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
Éléments syntaxiques appartenant à la procédure de codage elle-même (description des tâches)							
<DecodeIPM-SubFunction>	B	m	I	SUBFUNC	"DecodeIPM"	–	
<Message>	B	m	I	MESSAGE	<Path-parameter>	–	Trajet vers le fichier contenant le message de personne à personne à décoder
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
En présence d'une seule partie corps							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert de la partie corps décodée
<Document>	B	m	O	FILENAME	<File-parameter>	–	Le nom de fichier peut être fixé à l'avance sur demande par l'application locale. En cas de réception d'une seule partie corps, le nom doit être conservé par l'application de communication. En cas de réception de plusieurs parties corps, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Spécifie le type de document reçu
En présence de plusieurs parties corps							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière
Éléments syntaxiques recevant l'information d'en-tête du message de personne à personne à décoder							
<IpmId>	B	m	O	IPM-ID	<Ipm-id-parameter>	–	Identificateur de message
<R-RecipientSpec>	B	m	I	RECIPIENT	"@" <File-of-r-recipient-spec>	–	Trajet du fichier contenant les spécifications Recipientspecs de tous les destinataires supposés

TABLEAU 107/T.611 (fin)

**Eléments syntaxiques supplémentaires associés à l'extension <DecodeIPMExtension>
pour les services de courrier électronique**

Elément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<R-OriginatorSpec>	B	m	I	ADDRESS	"@" <File-of-originatorspec>	–	Nom de trajet du fichier contenant les coordonnées de l'expéditeur et des utilisateurs qui donnent les visas d'expédition
<ContType>	B	o	O	CONT-TYPE	<Content-type-parameter>	–	Type de contenu
<ExpiryTime>	B	o	O	EXPIRYTIME	<Date-time-parameter>	–	Date-heure d'expiration du message
<Forwarded>	B	o	O	FORWARDED	<Boolean-parameter>	–	Indication retransmise automatiquement
<Importance>	B	o	O	IMPORTANCE	<Importance-parameter>	–	Importance du contenu
<Language>	B	o	O	LANGUAGE	<Language-id-parameter>	–	Identification de la langue utilisée
<Priority>	B	o	O	PRIORITY	<Priority-parameter>	–	Priorité du contenu
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-relatedspec>	–	Nom de trajet du fichier contenant les spécifications Relatedspecs de tous les messages liés au présent message
<ReplyId>	B	o	O	REPLYID	<Ipm-id-parameter>	–	ID du message auquel le présent message répond
<ReplyTime>	B	o	O	REPLYTIME	<Date-time-parameter>	–	Date-heure à laquelle le(s) destinataire(s) doit (doivent) répondre aux utilisateurs qui donnent les visas d'expédition
<Sensitivity>	B	o	O	SENSITIVITY	<Sensitivity-parameter>	"0"	Confidentialité du contenu
<Subject>	B	o	O	SUBJECT	<Subject-parameter>	–	
<Userinfo>	+	o	O	USERINFO	<Userinfo-parameter>	–	Fourni par l'utilisateur

16.4 Notification de personne à personne

Une application locale peut transmettre une notification de personne à personne (IPN) avec la description TDD Send et recevoir une notification de personne à personne avec la description TDD Receive.

La notification IPN est traitée par l'application de communication comme un document particulier, appelé document IPN, c'est-à-dire que l'application de communication attend de l'application locale un fichier spécialement préparé contenant les paramètres IPN, lorsqu'une notification IPN doit être envoyée. L'application de communication doit convertir le document IPN selon le format imposé par le système de courrier électronique et le transmettre au destinataire. Inversement, l'application de communication doit présenter un document IPN à l'application locale lorsqu'elle reçoit une notification par le réseau.

Pour pouvoir distinguer les messages IPM des notifications IPN (et des rapports de remise, voir 16.5), on utilise le paramètre Content-type de l'élément syntaxique <ContType> (voir aussi 16.2.2.9).

Le document IPN doit exactement se conformer aux mêmes règles syntaxiques que celles qui s'appliquent aux descriptions TDD.

La syntaxe et le codage de type texte des documents IPN sont décrits dans les paragraphes ci-après.

16.4.1 Syntaxe du document IPN

Le document IPN est défini pour les systèmes de messagerie et est conforme à la syntaxe de type BNF suivante (voir le Tableau 108).

```
<IPN-Document> := <NotifyType> <SubjectIPM> <ReceiptTime>
                  [<IPNOrigin>] [<PrefRecipient>] [<Reason>] [<AutoComment>] [<SupplmInfo>]
```

TABLEAU 108/T.611

Éléments syntaxiques associés au document IPN

Élément syntaxique	Objet
<AutoComment>	Commentaire de l'expéditeur de la notification, si le message IPM associé a été retransmis automatiquement; est uniquement utilisé pour les notifications de non-réception du type NRN-FORW
<Reason>	Précise pourquoi le message IPM associé à la notification a été mis au rebut: 0: ipm expiré 1: ipm périmé 2: l'utilisateur n'est plus abonné uniquement utilisé pour les notifications de non-réception du type NRN-DISC
<IPNOrigin>	Identifie l'expéditeur de la notification
<NotifyType>	Type de notification: RN: notification de réception NRN-DISC: notification de non-réception (mis au rebut) NRN-FORW: notification de non-réception (retransmission automatique) OTHER: autres types de notification
<PrefRecipient>	Destinataire "préféré" du message IPM associé à la notification
<ReceiptTime>	Identifie la date-heure à laquelle l'expéditeur de la notification a reçu le message IPM associé; est uniquement utilisé pour les notifications de réception (RN)
<SubjectIPM>	Identifie le message IPM associé à la notification
<SupplmInfo>	Informations supplémentaires de l'agent de transfert de messages expéditeur; est uniquement utilisé pour les notifications de réception (RN)

16.4.2 Codage de type texte

16.4.2.1 Correspondance des mots clés

Voir le Tableau 109.

TABLEAU 109/T.611

Codage de type texte des éléments syntaxiques IPN pour les services de courrier électronique

(Le signe ↵ est le caractère "ligne suivante")

Élément syntaxique	Couple mot clé/paramètre
<AutoComment>	"AUTO-COMMENT" ":" STRING (SIZE(1..256)) ↵
<Reason>	"REASON" ":" "0" "1" "2" ↵
<IPNOrigin>	"IPN-ORIGIN" ":" <Mhs-or-descriptor> ↵
<NotifyType>	"NOTIFY-TYPE" ":" "RN" "NRN-DISC" "NRN-FORW" "OTHER" ↵
<PrefRecipient>	"DISCLO-REC" ":" <Mhs-or-descriptor> ↵
<ReceiptTime>	"RECEIPT-TIME" ":" <Date-time-parameter> ↵
<SubjectIPM>	"SUBJECT-IPM" ":" "@" <Ipm-id-parameter> ↵
<SupplmInfo>	"SUPPLM-INFO" ":" STRING (SIZE(1..256)) ↵

16.4.2.2 Codage des paramètres

Pour la description du codage de type texte des paramètres:

- <Mhs-or-descriptor> voir 16.2.2.4
- <Date-time-parameter> voir 6.4.4.4
- <Ipm-id-parameter> voir 16.2.2.11

16.5 Rapport de remise

Une application locale peut recevoir un rapport de remise (DR) avec la description TDD Receive.

Le rapport de remise est traité par l'application de communication comme un document particulier, appelé document rapport de remise. Par ailleurs, l'application de communication doit présenter le document rapport de remise à l'application locale lorsqu'elle reçoit un rapport de remise par le réseau.

Pour pouvoir distinguer les messages IPM des rapports de remise (et des notifications IPN, voir 16.4), on utilise le paramètre Content-type de l'élément syntaxique <ContType> (voir aussi 16.2.2.9).

Le document rapport de remise remis par l'application de communication doit exactement se conformer aux mêmes règles syntaxiques que celles appliquées aux descriptions TDD.

La syntaxe et le codage de type texte des documents rapports de remise sont décrits dans les paragraphes ci-après.

16.5.1 Syntaxe des documents rapports de remise

Le document rapport de remise est défini pour les systèmes de messagerie et est conforme à la syntaxe de type BNF suivante (voir le Tableau 110).

```
<DR-Document> := <ReportType> <ActRecipient> <MsgSubId> [<ReceiveTime>] [<Reason>]
                  [<Diagnostic>] [<SupplmInfo>]
```

TABLEAU 110/T.611

Éléments syntaxiques du document rapport de remise

Élément syntaxique	Objet
<ActRecipient>	Identifie le destinataire réel du message IPM associé au rapport
<Diagnostic>	Information supplémentaire de diagnostic associée au rapport, telle que définie dans la Recommandation UIT-T X.411; est uniquement utilisé pour les rapports de non-remise (NDR)
<MsgSubId>	Identifie le message IPM associé au rapport (MTSIdentifier)
<Reason>	Indique la raison pour laquelle le message IPM associé au rapport n'a pas été remis tel que défini dans la Recommandation UIT-T X.411; est uniquement utilisé pour les rapports de non-remise (NDR)
<ReceiveTime>	Date-heure à laquelle le message a été remis; est uniquement utilisé pour les rapports de remise (DR)
<ReportType>	Type de rapport de remise: DR: rapport de remise NDR: rapport de non-remise
<SupplmInfo>	Informations supplémentaires fournies par l'agent de transfert de messages expéditeur

16.5.2 Codage de type texte**16.5.2.1 Correspondance des mots clés**

Voir le Tableau 111.

TABLEAU 111/T.611

Codage de type texte des éléments syntaxiques des rapports de remise pour les services de courrier électronique

(Le signe ↵ est le caractère "ligne suivante")

Élément syntaxique	Couple mot clé/paramètre
<ActRecipient>	"ACT-RECIPIENT" ":" <Mhs-or-name> ↵
<Diagnostic>	"DIAGNOSTIC" ":" NUMERIC-STRING ↵
<MsgSubId>	"MSG-SUB-ID" ":" <Msg-sub-id-parameter> ↵
<Reason>	"REASON" ":" NUMERIC-STRING ↵
<ReceiveTime>	"RCVTIME" ":" <Date-time-parameter> ↵
<ReportType>	"REPORT-TYPE" ":" "DR" "NDR" ↵
<SupplmInfo>	"SUPPLM-INFO" ":" STRING (SIZE(1..256)) ↵

16.5.2.2 Codage des paramètres

Pour la description du codage de type texte des paramètres:

- <Mhs-or-name> voir 16.2.2.4
- <Msg-sub-id-parameter> voir 16.2.2.13
- <Date-time-parameter> voir 6.4.4.4

16.6 Correspondance avec éléments de service des systèmes de messagerie

Le Tableau 112 donne la correspondance entre les éléments de service des systèmes de messagerie tels que définis dans les Recommandations UIT-T de la série X.400 et les éléments syntaxiques définis dans la présente Recommandation. La liste ci-dessous donne les mots clés correspondants utilisés dans la présentation de type texte des descriptions TDD et précise leur prise en charge par les applications locales émettrices ou réceptrices.

TABLEAU 112/T.611

Correspondance des éléments de service des systèmes de messagerie

Élément de service du système de messagerie	Élément syntaxique	Mot clé	Application locale émettrice	Application locale réceptrice
Indication: destinataires suppléants admis	<Alternate>	ALTERNATE	Pris en charge	Non pris en charge
Indication de visa d'expédition	<R-OriginatorSpec> <S-OriginatorSpec>	ADDRESS ORIGINATOR	– Pris en charge	Pris en charge –
Indication de message avec retransmission automatique	<Forwarded>	FORWARDED	Non pris en charge	Pris en charge
Indication de destinataires de copies muettes	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Indication du type de contenu	<ContType>	CONT-TYPE	Pris en charge	Pris en charge
Interdiction de conversion	<ImplicitConv>	IMPLICIT-CONV	Pris en charge	Non pris en charge
Indication de références croisées	<RelatedSpec>	RELATED	Pris en charge	Pris en charge
Indication de remise différée	<SendTime>	SENDTIME	Pris en charge	Non pris en charge
Demande de rapport de remise	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Indication de la date-heure de remise	<ReceiveTime>	RCVTIME	Non pris en charge	Pris en charge
Divulgateion des autres destinataires	<DiscloRec>	DISCLO-REC	Pris en charge	Non pris en charge
Indication de la date d'expiration	<ExpiryTime>	EXPIRYTIME	Pris en charge	Pris en charge
Message de personne à personne retransmis	<Type>	TYPE	Pris en charge	Pris en charge
Sélection du niveau de priorité de remise	<Priority>	PRIORITY	Pris en charge	Pris en charge
Indication d'importance	<Importance>	IMPORTANCE	Pris en charge	Pris en charge
Indication de copie incomplète ^{a)}	<IncompleteCopy>	INC-Copy	Non pris en charge	Pris en charge
Indication du destinataire désigné	<R-RecipientSpec>	RECIPIENT	Non pris en charge	Pris en charge
Indication de message de personne à personne	<IpmId>	IPM-ID	Pris en charge	Pris en charge

TABLEAU 112/T.611 (fin)

Correspondance des éléments de service des systèmes de messagerie

Élément de service du système de messagerie	Élément syntaxique	Mot clé	Application locale émettrice	Application locale réceptrice
Indication de langue	<Language>	LANGUAGE	Non pris en charge	Pris en charge
Indication de la date-heure limite de remise ^{a)}	<LastTime>	LASTTIME	Pris en charge	Non pris en charge
Identification du message ^{a)}	<Userinfo>	USERINFO	Pris en charge	Pris en charge
Identification de la soumission du message	<MsgSubId>	MSG-SUB-ID	Pris en charge ^{b)}	Non pris en charge
Remise multidestinataires	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Corps multiparties	Implicit	n/a ^{c)}	Pris en charge	Pris en charge
Notification de non-réception	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Rapport de non-remise	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Indication de message de personne à personne périmé	<RelatedSpec>	RELATED	Pris en charge	Pris en charge
Indication de l'expéditeur	<R-OriginatorSpec> <S-OriginatorSpec>	ADDRESS ORIGINATOR	– Pris en charge	Pris en charge –
Indication de destinataires principaux et des destinataires de copies	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Indication d'essai	<Convert>	CONVERT	Pris en charge	Non pris en charge
Notification de réception	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Indication de message de personne à personne de réponse	<ReplyId>	REPLYID	Pris en charge	Pris en charge
Indication de destinataires de réponse	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Pris en charge	Pris en charge –
Indication de message de personne à personne de réponse	<RelatedSpec>	RELATED	Pris en charge	Pris en charge
Indication de la date-heure de réponse	<ReplyTime>	REPLYTIME	Pris en charge	Pris en charge
Indication de confidentialité	<Sensitivity>	SENSITIVITY	Pris en charge	Pris en charge
Indication de l'objet	<Subject>	SUBJECT	Pris en charge	Pris en charge
Indication de la date-heure de soumission	<SubmitTime>	SUBMITTIME	Pris en charge ^{b)}	Pris en charge
<p>a) Les éléments de service IncompleteCopyIndication, LatestDeliveryIndication et MessageIdentification étant des éléments supplémentaires, ils ne sont pas concernés par la définition du profil.</p> <p>b) Les entités MessageSubmissionIdentification et SubmissionTimeStamp sont produites et actualisées par le système de transfert de messages (MTS). Ces deux mots clés sont utilisés dans la fonction <SendAckTDD> comme paramètre de sortie.</p> <p>c) N'est pas concerné par l'attribution d'un mot clé.</p>				

Le Tableau 113 donne un récapitulatif des oblitérations date-heure utilisées dans la présente Recommandation et associées aux éléments de service du système de transfert de messages et du système de messagerie de personne à personne.

Ces éléments de service, leur signification et les mots clés correspondants du codage de type texte sont indiqués dans le Tableau 113. Une autre colonne précise si les indications temporelles peuvent être spécifiées par l'application locale expéditrice (Send TDD) et si elles peuvent être reçues par une application locale (Receive TDD).

TABLEAU 113/T.611

Récapitulatif des oblitérations date-heure pour les systèmes de messagerie

Élément de service	Mot clé	Envoi/réception	Signification
Indication de date d'expiration	EXPIRYTIME	e/r	Ce champ d'en-tête facultatif d'un message IPM spécifie la date à laquelle le ou les utilisateurs qui donnent les visas d'expédition considèrent que le message IPM a perdu sa validité
Indication de date-heure de réponse	REPLYTIME	e/r	Ce champ d'en-tête facultatif d'un message IPM spécifie la date-heure à laquelle le ou les utilisateurs qui donnent les visas d'expédition demandent que toutes les réponses au message IPM en question soient produites
Indication de remise différée	SENDTIME	e/-	Spécifie la date-heure avant laquelle le message ne doit pas être remis au(x) destinataire(s); il peut être produit par l'expéditeur du message
Indication de dernière remise	LASTTIME	e/-	Spécifie la date-heure après laquelle le message ne doit pas être remis au(x) destinataire(s); il peut être produit par l'expéditeur du message
Indication de la date-heure de remise	RCVTIME	-/r	Spécifie la date-heure à laquelle le message a été remis à l'utilisateur du système de transfert de messages destinataire; il doit être produit par le système de transfert de messages si le message a été effectivement remis
Indication de la date-heure de soumission	SUBMITTIME	e/r	Spécifie la date-heure à laquelle le système de transfert de messages accepte la responsabilité du message; il doit être produit par le système de transfert de messages
Indication de la date-heure de réception	RECEIPT-TIME	e/r	Uniquement utilisé pour la notification de réception (RN); identifie la date-heure à laquelle l'expéditeur d'une notification a reçu le message IPM associé

16.7 Profils du courrier électronique

La présente Recommandation définit l'accès aux services de courrier électronique décrits dans les Recommandations UIT-T de la série X.400 ainsi qu'aux autres environnements de courrier électronique.

La spécification d'une interface est devenue nécessaire compte tenu du développement de puissants services de télécommunication avec enregistrement et retransmission. Les fonctions TDD Send, SendAck et Receive ont été améliorées par la définition d'éléments de base propres au service de courrier électronique conformes aux principes énoncés dans la présente Recommandation. La plupart de ces éléments de service sont facultatifs.

Les nouvelles possibilités offertes au niveau de cette interface pour l'accès au courrier électronique sont les suivantes:

- transmission et réception de notifications;
- réception de rapports de remise;
- transmission et réception de messages de personne à personne intégrés dans des messages de courrier électronique (par exemple partie corps d'un message IPM).

Il faut pour cela qu'une mise en œuvre particulière d'une application de communication soit conforme à la présente Recommandation, même si toutes les caractéristiques possibles ne sont pas prises en charge. Pour améliorer les possibilités d'échange d'informations entre les applications locales et les applications de communication des différents fournisseurs, on a introduit le concept de "profil".

Les profils:

- X400-IPM;
- BasicX400-IPM;
- SimpleEMail

définissent les sous-ensembles fonctionnels du service présents au niveau de l'interface décrite dans la présente Recommandation¹⁵⁾.

Le concept de profil, associé à l'environnement de configuration d'interface (ICE) et aux classes fonctionnelles, identifie les éléments du service de courrier électronique pris en charge par une application de communication.

Une application de communication peut accepter un ou plusieurs profils. De plus, les ingénieurs d'application peuvent décrire leurs propres profils et leur donner un nom.

16.7.1 Profil de service X400-IPM

Le profil X400-IPM comprend tous les éléments du service X.400 présents au niveau de l'interface décrite dans la présente Recommandation pour la prise en charge de l'acheminement des messages de personne à personne¹⁶⁾ (voir le Tableau 114).

16.7.2 Profil de service BasicX400-IPM

Le profil BasicX400-IPM spécifie le sous-ensemble élémentaire des éléments du service X.400 présents au niveau de l'interface décrite dans la présente Recommandation pour la prise en charge de l'acheminement des messages de personne à personne (voir le Tableau 115).

16.7.3 Profil de service SimpleEMail

Le profil SimpleEMail a été créé pour permettre l'accès aux services de courrier électronique non conformes aux Recommandations UIT-T de la série X.400 sur les communications ouvertes. Il définit un sous-ensemble fonctionnel minimal d'éléments de services de courrier électronique présents au niveau de l'interface spécifiée dans la présente Recommandation (voir le Tableau 116).

16.8 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Une application de communication prenant en charge les services de courrier électronique doit spécifier dans le descripteur CA-Descriptor les parties corps (formats de transmission), le ou les profils et les types de contenu pris en charge (voir aussi 9.5).

La prise en charge des descripteurs TDD EncodeIPM et DecodeIPM, impose un choix approprié du mot clé EXTEND du descripteur CA-Descriptor (voir le Tableau 117).

17 Service: Transfert de fichiers

L'interface spécifiée dans la présente Recommandation permet l'envoi et la réception d'informations au moyen des services de transfert de fichiers. Cette interface n'a pas pour objet d'assurer une gestion complète des services de transfert de fichiers.

¹⁵⁾ Les autres profils comme les profils EDIM et EDIN appellent un complément d'étude.

¹⁶⁾ Le profil X400-IPM est spécifié conformément au manuel EPHOS (European Procurement Handbook for Open Systems), à savoir les profils ENV 41201 et ENV 41202.

TABLEAU 114/T.611

Profil de service X400-IPM

Élément de service	Élément de syntaxe	Prise en charge de l'application de communication
Indication: destinataires suppléants admis	<Alternate>	Obligatoire
Indication de visa d'expédition	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatoire
Indication de message avec retransmission automatique	<Forwarded>	Obligatoire
Indication de destinataires de copies muettes	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication du type de contenu	<ContType>	Obligatoire
Interdiction de conversion	<ImplicitConv>	Obligatoire
Indication de références croisées	<RelatedSpec>	Obligatoire
Indication de remise différée	<SendTime>	Obligatoire
Demande de rapport de remise	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication de la date-heure de remise	<ReceiveTime>	Obligatoire
Divulgarion des autres destinataires	<DiscloRec>	Obligatoire
Indication de la date d'expiration	<ExpiryTime>	Obligatoire
Message de personne à personne retransmis	<Type>	Obligatoire
Sélection du niveau de priorité de remise	<Priority>	Obligatoire
Indication d'importance	<Importance>	Obligatoire
Indication du destinataire désigné	<R-RecipientSpec>	Obligatoire
Indication de message de personne à personne	<IpmId>	Obligatoire
Indication de langue	<Language>	Obligatoire
Identification de la soumission du message	<MsgSubId>	Obligatoire
Remise multidestinations	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Corps multiparties	Implicit	Obligatoire
Notification de non-réception	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Rapport de non-remise	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication de message de personne à personne périmé	<RelatedSpec>	Obligatoire
Indication d'expéditeur	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatoire
Indication de destinataires principaux et des destinataires de copies	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication d'essai	<Convert>	Obligatoire
Notification de réception	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication de message de personne à personne de réponse	<ReplyId>	Obligatoire
Indication de destinataires de réponses	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication de message de personne à personne de réponse	<RelatedSpec>	Obligatoire
Indication de la date-heure de réponse	<ReplyTime>	Obligatoire
Indication de confidentialité	<Sensitivity>	Obligatoire
Indication de l'objet	<Subject>	Obligatoire
Indication de la date-heure de soumission	<SubmitTime>	Obligatoire

TABLEAU 115/T.611

Profil de service BasicX400-IPM

Élément de service	Élément de syntaxe	Prise en charge de l'application de communication
Indication: destinataires suppléants admis	<Alternate>	–
Indication de visa d'expédition	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatoire
Indication de message avec retransmission automatique	<Forwarded>	–
Indication de destinataires de copies muettes	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication du type de contenu	<ContType>	Obligatoire
Interdiction de conversion	<ImplicitConv>	–
Indication de références croisées	<RelatedSpec>	Obligatoire
Indication de remise différée	<SendTime>	Obligatoire
Rapport de remise	<R-RecipientSpec>, <S-RecipientSpec>	–
Indication de la date-heure de remise	<ReceiveTime>	–
Divulgaration des autres destinataires	<DiscloRec>	–
Indication de la date d'expiration	<ExpiryTime>	Obligatoire
Message de personne à personne retransmis	<Type>	Obligatoire
Sélection du niveau de priorité de remise	<Priority>	Obligatoire
Indication d'importance	<Importance>	Obligatoire
Indication du destinataire désigné	<R-RecipientSpec>	–
Indication de message de personne à personne	<IpmId>	Obligatoire
Indication de langue	<Language>	Obligatoire
Identification de la soumission du message	<MsgSubId>	–
Remise multidestinations	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Corps multiparties	Implicit	Obligatoire
Notification de non-réception	<R-RecipientSpec>, <S-RecipientSpec>	–
Rapport de non-remise	<R-RecipientSpec>, <S-RecipientSpec>	–
Indication de message de personne à personne périmé	<RelatedSpec>	Obligatoire
Indication d'expéditeur	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatoire
Indication de destinataires principaux et des destinataires de copies	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication d'essai	<Convert>	–
Notification de réception	<R-RecipientSpec>, <S-RecipientSpec>	–
Indication de message de personne à personne de réponse	<ReplyId>	Obligatoire
Indication de destinataires de réponses	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication de message de personne à personne de réponse	<RelatedSpec>	Obligatoire
Indication de la date-heure de réponse	<ReplyTime>	Obligatoire
Indication de confidentialité	<Sensitivity>	Obligatoire
Indication de l'objet	<Subject>	Obligatoire
Indication de la date-heure de soumission	<SubmitTime>	–

TABLEAU 116/T.611

Profil de service SimpleEMail

Élément de service	Élément de syntaxe	Prise en charge de l'application de communication
Indication: destinataires suppléants admis	<Alternate>	–
Indication de visa d'expédition	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatoire
Indication de message avec retransmission automatique	<Forwarded>	–
Indication des destinataires de copies muettes	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication du type de contenu	<ContType>	Obligatoire
Interdiction de conversion	<ImplicitConv>	–
Indication de références croisées	<RelatedSpec>	Obligatoire
Indication de remise différée	<SendTime>	–
Rapport de remise	<R-RecipientSpec>, <S-RecipientSpec>	–
Indication de la date-heure de remise	<ReceiveTime>	–
Divulgateion des autres destinataires	<DiscloRec>	–
Indication de la date d'expiration	<ExpiryTime>	–
Message de personne à personne retransmis	<Type>	–
Sélection du niveau de priorité de remise	<Priority>	Obligatoire
Indication d'importance	<Importance>	–
Indication du destinataire désigné	<R-RecipientSpec>	–
Indication de message de personne à personne	<IpmId>	Obligatoire
Indication de langue	<Language>	–
Identification de la soumission du message	<MsgSubId>	–
Remise multidestinations	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Corps multiparties	Implicit	Obligatoire
Notification de non-réception	<R-RecipientSpec>, <S-RecipientSpec>	–
Rapport de non-remise	<R-RecipientSpec>, <S-RecipientSpec>	–
Indication de message de personne à personne périmé	<RelatedSpec>	–
Indication d'expéditeur	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatoire
Indication de destinataires principaux et des destinataires de copies	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication d'essai	<Convert>	–
Notification de réception	<R-RecipientSpec>, <S-RecipientSpec>	–
Indication de message de personne à personne de réponse	<ReplyId>	Obligatoire
Indication de destinataires de réponses	<R-RecipientSpec>, <S-RecipientSpec>	Obligatoire
Indication de message de personne à personne de réponse	<RelatedSpec>	Obligatoire
Indication de la date-heure de réponse	<ReplyTime>	Obligatoire
Indication de confidentialité	<Sensitivity>	Obligatoire
Indication de l'objet	<Subject>	Obligatoire
Indication de la date-heure de soumission	<SubmitTime>	–

TABLEAU 117/T.611

Valeurs supplémentaires du descripteur CA-Descriptor pour les services de courrier électronique

Mot clé	Paramètre	L'application de communication déclare que
EMAIL	"STD"	les parties corps textuelles sont prises en charge; "STD" établit la correspondance avec l'alphabet de base approprié du service de courrier électronique sélectionné
EMAIL	"TELETEX"	les parties corps du télétexte sont prises en charge
EMAIL	"VIDEOTEX"	les parties corps du vidéotex sont prises en charge
EMAIL	"G3FAX"	les parties corps de la télécopie du groupe 3 sont prises en charge
EMAIL	"G4CLASS1"	les parties corps de la télécopie du groupe 4 sont prises en charge
EMAIL	"MIXEDMODE"	les parties corps du mode mixte sont prises en charge
EMAIL	"MESSAGE"	les parties corps de la messagerie IPM sont prises en charge
EMAIL	"BILATERAL" ^{a)}	les parties corps définies bilatéralement sont prises en charge
EMAIL	"NATIONAL"	les parties corps définies nationalement sont prises en charge
EMAIL	"ODA"	les parties corps ODA sont prises en charge
CONT-TYPE	"mhsIPMxx"	le protocole du système de messagerie applicable au contenu IPM est pris en charge (xx désigne l'année de publication de la Recommandation associée: par exemple 84 pour 1984)
CONT-TYPE	"mhsIPNxx"	le protocole du système de messagerie applicable au contenu IPN est pris en charge (pour l'application de communication il s'agit d'un document IPN au sens du 16.4)
CONT-TYPE	"mhsDR"	le protocole du système de messagerie applicable au contenu rapport de remise est pris en charge (pour l'application de communication il s'agit d'un rapport de remise au sens du 16.5)
PROFILE	"X400-IPM"	le profil X.400 est pris en charge
PROFILE	"BasicX400-IPM"	le profil BasicX400 est pris en charge
PROFILE	"SimpleEMail"	le profil SimpleEMail est pris en charge
EXTEND	"EncodeIPM"	la fonctionnalité EncodeIPM est prise en charge
EXTEND	"DecodeIPM"	la fonctionnalité DecodeIPM est prise en charge
<p>^{a)} Le terme "Bilateral" est utilisé pour d'autres types de partie corps de texte non mentionnés dans la liste.</p> <p>NOTE – Une application de communication peut également prendre en charge d'autres systèmes de courrier électronique publics ou privés. Dans ce cas, le type du système de courrier électronique doit apparaître dans les paramètres CONT-TYPE et PROFILE (par exemple "CONT-TYPE : cmcIPM" pour la prise en charge des interfaces CMC, voir aussi 16.2.2.9).</p>		

17.1 Eléments syntaxiques propres au service

```

<ServiceDependentKeywordsSend> :=
    (<Recipient> | <RecipientSpec>)
    (((<Document> <Convert> [<Type>]) | <DocumentSpec>)
    <Compress> <Environ> [<Password>] [<Name>])

```

```

<ServiceDependentKeywordsSendAck> :=
    <Recipient>
    (((<Document> <Convert> [<Type>]) | <DocumentSpec>)
    <Compress> <Environ> [<Password>] [<Name>])

```

<ServiceDependentKeywordsReceive> :=
 [<Originator>]
 ((<Document> <Convert> <Type>) | <DocumentSpec>)
 <Compress> <Environ> [<Password>] [<Name>]

Voir le Tableau 118.

TABLEAU 118/T.611

Éléments syntaxiques supplémentaires pour les services de transfert de fichiers

Elément syntaxique	Objet
<Compress>	Spécifie le format de compression de données du ou des fichiers transférés
<Convert>	Spécifie le format de transfert à utiliser
<Document>, <DocumentSpec>	Spécifie le(s) document(s) à envoyer ou en cours de réception
<Environ>	Spécifie le système d'exploitation sous lequel le fichier a été créé par l'expéditeur
<Name>	Nom du fichier pour le système d'exploitation de l'expéditeur
<Originator>	Spécifie l'adresse de communication de l'expéditeur
<Password>	Mot de passe nécessaire pour avoir accès à la station distante
<Recipient>, <RecipientSpec>	Spécifie la ou les adresses de communication du ou des destinataires
<Type>	Spécifie le format de transmission utilisé

17.2 Codage de type texte

17.2.1 Correspondance des mots clés

Voir le Tableau 119.

TABLEAU 119/T.611

Codage de type texte des éléments syntaxiques pour les services de transfert de fichiers

(Le signe ↵ désigne le caractère "ligne suivante")

Elément syntaxique	Couple mot clé/paramètre
<Compress>	"COMPRESS" ":" <Compress-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<Environ>	"ENVIRON" ":" <Environ-parameter> ↵
<Name>	"NAME" ":" <Name-parameter> ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Password>	"PASSWORD" ":" <Password-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵

17.2.2 Codage des paramètres

Voir aussi 6.4.4 pour le codage des paramètres non dépendants du service utilisés.

17.2.2.1 Paramètre Service-id

Le paramètre Service-id est codé comme une chaîne STRING dont la valeur est constante et égale à "FT".

Syntaxe:

```
<Service-id-parameter> := "FT"
```

17.2.2.2 Paramètre Type-id

Le paramètre Type-id est codé comme une chaîne STRING dont la valeur est constante et égale à "STD".

Syntaxe:

```
<Type-id-parameter> := "STD"
```

17.2.2.3 Paramètre Convert-id

Le paramètre Convert-id est codé comme une chaîne STRING dont la valeur est constante et égale à "VOID".

Syntaxe:

```
<Convert-id-parameter> := "VOID"
```

17.2.2.4 Paramètres File-of-addrspec et Address

Le paramètre File-of-addrspec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Addrspec, qui contiennent le paramètre Address.

Le paramètre Address est codé comme une chaîne STRING. La chaîne STRING contient le numéro de téléphone. Si ce numéro de téléphone commence par un point d'exclamation ("!"), il peut contenir des caractères spéciaux qui sont traités comme des opérateurs (ou des modificateurs) et non comme des chiffres. Les caractères autorisés appellent un complément d'étude.

On peut aussi donner un nom de substitution à la place du numéro de téléphone, mais il faut alors que ce nom soit précédé du caractère "&". On suppose que l'application de communication sait décoder le nom de substitution spécifié.

Syntaxe:

```
<File-of-addrspec> := PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Addrsec-parameter>
```

```
<Addrsec-parameter> := <Address-parameter>
```

```
<Address-parameter> := <Phone-number> | ("!" <Dial-command>) | ("&" <Alias>)
-- contient le numéro de téléphone, la séquence de numérotation ou le
-- nom de substitution
```

```
<Phone-number> := NUMERIC-STRING
```

```
<Dial-command> := STRING
-- appelle un complément d'étude
```

```
<Alias> := STRING
```

17.2.2.5 Paramètres File-of-filespec et File

Le paramètre File-of-filespec est codé comme un trajet PATH pointant sur un fichier contenant les paramètres Filespec, qui contiennent le paramètre File. Le paramètre File lui-même est codé comme un trajet PATH pointant sur le fichier transféré.

Syntaxe:

```
<File-of-filespec> :=          PATH
-- Le trajet pointe sur un fichier contenant un ou plusieurs paramètres
-- <Filespec-parameter>

<Filespec-parameter> :=      <File-parameter> "," <File-conv>
["," [<File-type>] ["," [<File-compress>] ["," <File-name>] ] ]

<File-parameter> :=          PATH
-- Trajet vers le fichier transféré

<File-conv> :=                <Convert-id-parameter>
-- Spécifie le format de transfert du fichier

<File-type> :=                <Type-id-parameter>
-- Spécifie le format de transmission du fichier

<File-compress> :=           <Compress-parameter>
-- Spécifie la compression

<File-name> :=                <Name-parameter>
-- Spécifie le nom du fichier
```

17.2.2.6 Paramètre Compress

Le paramètre Compress est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"VOID"	Pas de compression (par défaut)
autre	Identificateur de compression

Syntaxe:

```
<Compress-parameter> :=      "VOID" | <Compress-identifiant>
<Compress-identifiant> :=     STRING
-- par exemple "ZIP", "AZJ", "ARJ", "LZH" ou "V42"
```

17.2.2.7 Paramètre Environ

Le paramètre Environ est codé comme une chaîne STRING prenant l'une des valeurs suivantes:

"MSDOS"	MS-DOS
"WINDOWS"	Windows
"UNIX"	Unix
"OS2"	OS/2
"MacOS"	Mac
autre	Réservé pour complément d'étude

Syntaxe:

```
<Environ-parameter> :=      "MSDOS" | "WINDOWS" | "UNIX" | "OS2" | "MacOS"
```

17.2.2.8 Paramètre Password

Le paramètre Password est codé comme une chaîne STRING.

Syntaxe:

```
<Password-parameter> :=     STRING
```

17.2.2.9 Paramètre Name

Le paramètre Name est codé comme une chaîne STRING comportant 1 à 16 caractères.

Syntaxe:

```
<Name-parameter> :=         STRING (SIZE(1..16))
```

17.3 Fonctionnalités additionnelles

17.3.1 Fonctions: Send et SendAck

Voir les Tableaux 120 et 121.

TABLEAU 120/T.611

Fonctionnalités additionnelles de la fonction <SendTDD> pour le transfert de fichiers

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Environ>	B	m	I	ENVIRON	<Environ-parameter>	–	Système d'exploitation de l'expéditeur
<Password>	B	o	I	PASSWORD	<Password-parameter>	–	Mot de passe pour l'accès au système de fichiers distant
Envoi à un destinataire							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
Envoi à un ou à plusieurs destinataires							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Spécifie une liste de destinataires
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique sortant à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<Compress>	B	o	I	COMPRESS	<Compress-parameter>	"VOID"	Compression
<Name>	B	o	I	NAME	<Name-parameter>	–	Nom du fichier dans l'environnement d'origine
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

TABLEAU 121/T.611

Fonctionnalités additionnelles de la fonction <SendAckTDD> pour le transfert de fichiers

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Recipient>	B	m	I/O	ADDRESS	<Address-parameter>	–	Spécifie le numéro d'appel d'un destinataire
<Environ>	B	m	I	ENVIRON	<Environ-parameter>	–	Système d'exploitation de l'expéditeur
<Password>	B	o	I	PASSWORD	<Password-parameter>	–	Mot de passe pour l'accès au système de fichiers distant
Envoi d'un seul fichier							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indique le format de transfert du fichier sortant
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichier unique sortant à transmettre, remis dans le format de transfert spécifié par le mot clé Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Spécifie le type de document à envoyer
<Compress>	B	o	I	COMPRESS	<Compress-parameter>	"VOID"	Compression
<Name>	B	o	I	NAME	<Name-parameter>	–	Nom du fichier dans l'environnement d'origine
Envoi d'un ou de plusieurs fichiers							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Spécifie une liste de fichiers; utilise une syntaxe particulière

17.3.2 Fonction: Receive

Voir le Tableau 122.

TABLEAU 122/T.611

Fonctionnalités additionnelles de la fonction <ReceiveTDD> pour le transfert de fichiers

Élément syntaxique	C	T	I/O	Codage de type texte			Observations
				Mot clé	Paramètre	Valeur par défaut	
<Environ>	B	m	O	ENVIRON	<Environ-parameter>	-	Système d'exploitation de l'expéditeur
<Password>	B	o	I	PASSWORD	<Password-parameter>	-	Mot de passe pour l'accès au système de fichiers distant
<Originator>	B	o	O	ADDRESS	<Address-parameter>	-	Spécifie le numéro de téléphone de l'expéditeur
Réception d'un seul fichier							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	-	Indique le format de transfert du fichier reçu
<Document>	B	m	I/O	FILENAME	<File-parameter>	-	Le nom de fichier peut être fixé à l'avance par l'application locale. En cas de réception d'un seul fichier, le trajet doit être conservé par l'application de communication. En cas de réception de plusieurs fichiers, le nom peut être "écrasé" par l'application de communication
<Type>	B	o	O	TYPE	<Type-id-parameter>	-	Spécifie le type de document reçu
<Compress>	B	o	O	COMPRESS	<Compress-parameter>	"VOID"	Compression
<Name>	B	o	O	NAME	<Name-parameter>	-	Nom du fichier dans l'environnement d'origine
Réception de plusieurs fichiers							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	-	Spécifie une liste de fichiers; utilise une syntaxe particulière

17.4 Valeurs du descripteur de l'application de communication (CA-Descriptor)

Une application de communication prenant en charge le service de transfert de fichiers doit spécifier le paramètre Type-id "STD" (voir le Tableau 123).

TABLEAU 123/T.611

Valeur supplémentaire du descripteur CA-Descriptor pour le transfert de fichiers

Mot clé	Paramètre	L'application de communication déclare que
FT	"STD"	le service de transfert de fichiers de base est assuré

PARTIE III – BINARY ENCODING SCHEME (SYSTÈME DE CODAGE BINAIRE)

18 Description générique en langage C

Le présent article décrit le système de codage binaire des descriptions TDD utilisant une description «générique» en langage C (voir aussi A.2). Le système de codage binaire exige que l'identificateur de code (code-id) soit mis sur «C» (voir 6.3). Cette description a pour but d'assurer la compatibilité binaire des descriptions TDD entre les applications des différents fournisseurs utilisant le système de codage binaire.

Les descriptions présentées ont été adaptées à la plate-forme de prise en charge pour qu'elles fonctionnent convenablement (voir aussi la Partie IV de la présente Recommandation).

Les types de données génériques en langage C ci-dessous sont utilisés pour décrire les descriptions TDD. Ces types de données doivent être adaptés à leurs types de données homologues en langage C dépendant de la plate-forme (voir le Tableau 124).

TABLEAU 124/T.611

Types de données génériques en langage C

Type de données génériques en langage C	Observation
INT16	Valeur d'entier avec signe, codée sur 16 bits
UINT32	Valeur d'entier sans signe, codée sur 32 bits
CHAR	Caractère, codé sur 8 bits
BYTE	Octet, codé sur 8 bits

18.1 Codage binaire des description TDD

Les descriptions TDD de demande et les réponses sont représentées par des structures en langage C. Les composants de ces structures correspondent aux éléments de syntaxe utilisés dans les formes de codage en mode texte, comme indiqué dans l'article 6 et dans les articles appropriés de la Partie II de la présente Recommandation.

La structure binaire, décrite à l'aide des structures en langage C a été choisie de manière que les éléments des différents types de données soient alignés sur une limite de 4 octets.

Chaque description TDD comprend successivement:

- un en-tête binaire de 4 octets;
- un descripteur TDD de 16 octets, avec les décalages correspondant aux quatre sections spécifiques de la structure TDD; ces décalages sont calculés à partir du point de départ de l'en-tête binaire.

Le reste de la structure binaire est constitué par les quatre sections spécifiques ci-dessous, qui peuvent se présenter dans n'importe quel ordre:

- *Section 1* – Contient les paramètres d'entrée communs, c'est-à-dire ceux qui sont décrits dans l'article 6.
- *Section 2* – Contient les paramètres d'entrée dépendant du service, c'est-à-dire ceux qui sont décrits dans l'article approprié de la Partie II de la présente Recommandation.
- *Section 3* – Contient les paramètres de sortie communs, c'est-à-dire ceux qui sont décrits dans l'article 6.
- *Section 4* – Contient les paramètres de sortie communs, c'est-à-dire ceux qui sont décrits dans l'article approprié de la Partie II de la présente Recommandation.

Pour certaines applications, cette structure permet de compléter ou d'améliorer les paramètres d'une description TDD à l'aide de paramètres privés.

Cette structure est représentée à la Figure 16.

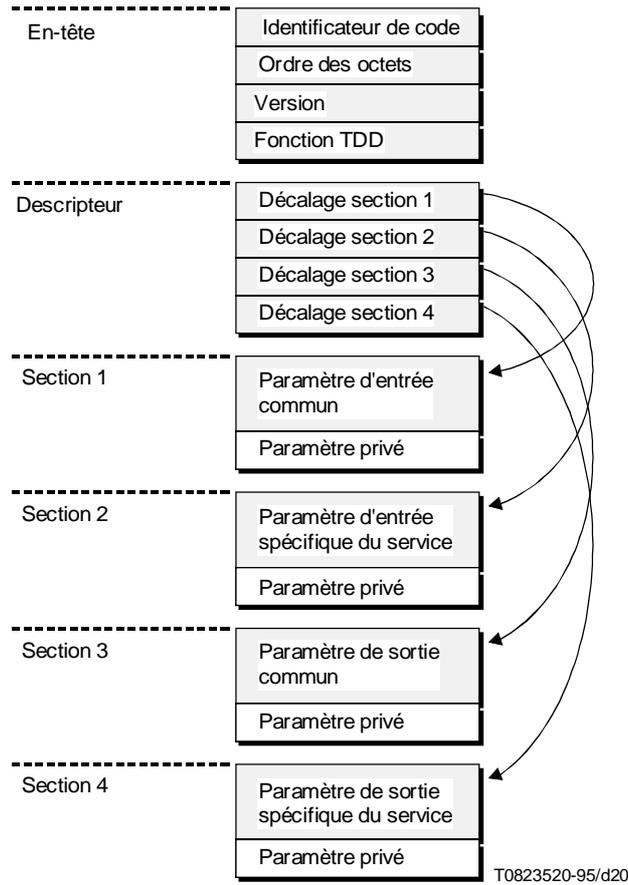


FIGURE 16/T.611
Structure binaire de TDD

18.1.1 En-tête TDD

```

/**
 * <TDD> syntax element
 ***/

```

```

struct tdd {
    struct tdd_header    hdr;           /* <TDD Header> syntax element, see below */
    struct tdd_descriptor ofs;         /* Descriptor containing offsets, see below */
};                                     /* Function dependent TDD structures ... */

```

```

/**
 * <TDD Header> syntax element
 ***/

```

```

struct tdd_header {
    BYTE_code_id;           /* Code-ID, constant set to 0x43 ('C') */
    BYTE_byte_order;       /* Optional constant value indicating byte order */
    BYTE_itu_version;      /* Constant indicating version of this Recommendation */
    BYTE_function;         /* Function code */
};

```

```

/**
 *      Offset descriptor
 ***/

```

```

struct tdd_descriptor {
    UINT32 cinp;           /* offset to section 1 (common input parameter) */
    UINT32 sinp;           /* offset to section 2 (service dependent input parameter) */
    UINT32 coup;          /* offset to section 3 (common output parameter) */
    UINT32 soup;          /* offset to section 4 (service dependent output parameter) */
};

```

18.1.2 Structures TDD indépendantes du service

```

/**
 *      <SendTDD> structure
 ***/

```

```

struct send_tdd_inp {
    La_id_type           la_id;           /* common input parms */
    Req_id_type          req_id;          /* <LaId> */
    Service_type         service;         /* <ReqID> */
    Date_time_type       sendtime;        /* <Service> */
    Date_time_type       lasttime;        /* <SendTime> */
    Comment_type         comment;         /* <LastTime> */
    Userkey_type         userkey;         /* <Comment> */
};

```

```

/**
 *      <SendackTDD> structure
 ***/

```

```

struct sendack_tdd_inp {
    La_id_type           la_id;           /* common input parms */
    Req_id_type          req_id;          /* <LaId> */
    Service_type         service;         /* <ReqID> */
    Date_time_type       sendtime;        /* <Service> */
    Date_time_type       lasttime;        /* <SendTime> */
    Comment_type         comment;         /* <LastTime> */
    Userkey_type         userkey;         /* <Comment> */
};

```

```

struct sendack_tdd_out {
    Comid_type           comid;           /* common output parms */
    Status_type          status;          /* <ComId> */
    Error_type           error;           /* <Status> */
    Error_type           minor;           /* <Error> */
    Error_type           warning;         /* <Minor> */
};

```

```

/**
 *      <ReceiveTDD> structure
 ***/

```

```

struct receive_tdd_inp {
    La_id_type           la_id;           /* common input parms */
    Req_id_type          req_id;          /* <LaId> */
    Service_type         service;         /* <ReqID> */
    Boolean_type         deletedoc;       /* <Service> */
    Comid_type           comid;           /* <Delete > */
};

```

```

struct receive_tdd_out {
    Comid_type      comid;      /* common output parms */
    Status_type     status;     /* <ComId> */
    Error_type      error;      /* <Status> */
    Error_type      minor;      /* <Error> */
    Error_type      warning;    /* <Minor> */
    Service_type    service;    /* <Warning> */
    Date_time_type  rcvtime;    /* <Service> */
    };                          /* <ReceiveTime> */

```

```

/****
*      <CopyTDD> structure
****/

```

```

struct copy_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <LaId> */
    Comid_type      comid;     /* <ReqID> */
    Req_id_type     reqref;    /* <ComId> */
    State_type      state;     /* <ReqRef> */
    Path_type       target;    /* <State> */
    Layout_id_type  layout;    /* <Target> */
    };                          /* <Layout> */

```

```

struct copy_tdd_out {
    Error_type      error;      /* common output parms */
    Error_type      minor;     /* <Error> */
    Error_type      warning;    /* <Minor> */
    };                          /* <Warning> */

```

```

/****
*      <DeleteTDD> structure
****/

```

```

struct delete_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <LaId> */
    Comid_type      comid;     /* <ReqID> */
    Req_id_type     reqref;    /* <ComId> */
    };                          /* <ReqRef> */

```

```

struct delete_tdd_out {
    Error_type      error;      /* common output parms */
    Error_type      minor;     /* <Error> */
    Error_type      warning;    /* <Minor> */
    };                          /* <Warning> */

```

```

/****
*      <CancelTDD> structure
****/

```

```

struct cancel_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <LaId> */
    Comid_type      comid;     /* <ReqID> */
    Req_id_type     reqref;    /* <ComId> */
    };                          /* <ReqRef> */

```

```

struct cancel_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/**
 * <PurgeTDD> structure
 */

struct purge_tdd_inp {
    La_id_type      la_id;
    Req_id_type     req_id;
    Comid_type      comid;
    Req_id_type     reqref;
    State_type      state;
};

struct purge_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/**
 * <RescheduleTDD> structure
 */

struct reschedule_tdd_inp {
    La_id_type      la_id;
    Req_id_type     req_id;
    Comid_type      comid;
    Req_id_type     reqref;
    Address_type    address;
    Date_time_type  sendtime;
    Date_time_type  lasttime;
};

struct reschedule_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/**
 * <DispatchTDD> structure
 */

struct dispatch_tdd_inp {
    La_id_type      la_id;
    Req_id_type     req_id;
    Comid_type      comid;
    La_id_type      newla;
};

struct dispatch_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

```

```

/**
 * <PreviewTDD> structure
 ***/

struct preview_tdd_inp {
    La_id_type      la_id;          /* common input parms */
    Req_id_type     req_id;        /* <ReqID> */
    Comid_type      comid;        /* <ComId> */
    Path_type       target;       /* <Target> */
    Convert_id_type convert;      /* <Convert> */
};

struct preview_tdd_out {
    Error_type      error;        /* common output parms */
    Error_type      minor;       /* <Error> */
    Error_type      warning;     /* <Minor> */
    Error_type      warning;     /* <Warning> */
};

/**
 * <PrintTDD> structure
 ***/

struct print_tdd_inp {
    La_id_type      la_id;          /* common input parms */
    Req_id_type     req_id;        /* <ReqID> */
    Path_type       filename;     /* <FileName> */
    Convert_id_type informat;     /* <InFormat> */
    Printer_id_type printer;      /* <Printer> */
};

struct print_tdd_out {
    Error_type      error;        /* common output parms */
    Error_type      minor;       /* <Error> */
    Error_type      warning;     /* <Minor> */
    Error_type      warning;     /* <Warning> */
};

/**
 * <ConvertTDD> structure
 ***/

struct convert_tdd_inp {
    La_id_type      la_id;          /* common input parms */
    Req_id_type     req_id;        /* <ReqID> */
    Path_type       filename;     /* <FileName> */
    Path_type       target;       /* <Target> */
    Convert_id_type informat;     /* <InFormat> */
    Convert_id_type outformat;    /* <OutFormat> */
};

struct convert_tdd_out {
    Error_type      error;        /* common output parms */
    Error_type      minor;       /* <Error> */
    Error_type      warning;     /* <Minor> */
    Error_type      warning;     /* <Warning> */
};

/**
 * <CheckTDD> structure
 ***/

```

```

struct check_tdd_inp {
    La_id_type      la_id;
    Req_id_type     req_id;
    Path_type       filename;
    Convert_id_type check;
};

/* common input parms */
/* <LaId> */
/* <ReqID> */
/* <FileName> */
/* <Check> */

struct check_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/* common output parms */
/* <Error> */
/* <Minor> */
/* <Warning> */

/**
 * <ExtendTDD> structure
 ***/

struct extend_tdd_inp {
    Subfunc_type    subfunc;
    La_id_type      la_id;
    Req_id_type     req_id;
};

/* common input parms */
/* <SubFunction> */
/* <LaId> */
/* <ReqID> */

struct extend_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/* common output parms */
/* <Error> */
/* <Minor> */
/* <Warning> */

/**
 * <NationalTDD> structure
 ***/

struct national_tdd_inp {
    subfunc; /*
    La_id_type      la_id;
    Req_id_type     req_id;
    /*           ...
};

/* common input parms */
/* <NationalFunction>, for further study */
/* <LaId> */
/* <ReqID> */
<--- others are added here, for further study */

struct national_c_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
    /*           ...
};

/* common output parms */
/* <Error> */
/* <Minor> */
/* <Warning> */
<--- others are added here, for further study */

/**
 * <PrivateTDD> structure
 ***/

struct private_tdd_inp {
    Subfunc_type    subfunc;
    La_id_type      la_id;
    Req_id_type     req_id;
    /*           ...
};

/* common input parms */
/* <PrivateFunction>, for further study */
/* <LaId> */
/* <ReqID> */
<--- others are added here, for further study */

```

```

struct private_tdd_out {
    Error_type      error;      /* common output parms */
    Error_type      minor;     /* <Error> */
    Error_type      warning;   /* <Minor> */
    /*      ...      /* <Warning> */
    /*      ...      /* --- others are added here, for further study */
};

```

18.1.3 Définitions du type et constantes indépendantes du service

```

/**
 * Constants used for fields of struct tdd_header
 */

/* code_id field */
#define CODE_ID      0x43      /* Code-ID, set to 'C' */

/* byte_order field */
#define L_BYTEORDER  0x4C      /* Little Endian (Intel, low-high) byte order */
#define B_BYTEORDER  0x42      /* Big endian byte order */

/* itu_version field */
#define ITU_VERSION   94      /* Current version (1994) */

/* function field */
#define SENDACK_TDD   0x10      /* <SendAckFunction> */
#define SEND_TDD      0x11      /* <SendFunction> */
#define RECEIVE_TDD   0x20      /* <ReceiveFunction> */
#define COPY_TDD      0x30      /* <CopyFunction> */
#define DELETE_TDD    0x31      /* <DeleteFunction> */
#define CANCEL_TDD    0x32      /* <CancelFunction> */
#define PURGE_TDD     0x33      /* <PurgeFunction> */
#define RESCHEDULE_TDD 0x34      /* <RescheduleFunction> */
#define DISPATCH_TDD  0x35      /* <DispatchFunction> */
#define PREVIEW_TDD   0x36      /* <PreviewFunction> */
#define PRINT_TDD     0x40      /* <PrintFunction> */
#define CONVERT_TDD   0x41      /* <ConvertFunction> */
#define CHECK_TDD     0x42      /* <CheckFunction> */
#define EXTEND_TDD    0x50      /* <ExtendFunction> */
#define NATIONAL_TDD  0x60      /* <NationalFunction> */
#define PRIVATE_TDD   0x70      /* <PrivateFunction> */

/**
 * Constants used for struct extend_tdd_inp
 */

/* subfunction field */
#define POLL_FX3      0x0101    /* FX3: Poll */

#define ENCODE_IPM    0x2010    /* EMAIL: EncodeIPM */
#define DECODE_IPM    0x2011    /* EMAIL: DecodeIPM */

/**
 * Type definitions
 * Some types are service dependent. Their values might be restricted or expanded upon further.
 */

```

```

/*
**      Address_type
**      Some telecommunication services may constrain or expand this type furthermore.
*/

typedef CHAR Address_type[127+1];

/*
**      Boolean_type
**      Takes the values true or false
*/

typedef enum { false=0, true } Boolean_type;

/*
**      Comid type
*/

typedef CHAR Comid_type[31+1];

/*
**      Comment type
*/

typedef CHAR Comment_type[127+1];

/*
**      Convert_id_type
**      The (asciiz) strings assigned for text based encoding shall be used.
*/

typedef CHAR Convert_id_type[15+1];

/*
**      Date_time_type and Send_time_type
**      The date and time specification follows closely the ANSI specification
**      The values "IMMEDIATE" and "URGENT" are set by specifying the seconds
**      part of the struct below (tm_sec) with (-1) or (-2)
*/

typedef struct {
    INT16          tm_sec;          /* seconds; used for "IMMEDIATE" and "URGENT" specification
    */
    INT16          tm_min;          /* minutes */
    INT16          tm_hour;         /* hours */
    INT16          tm_mday;         /* day of month */
    INT16          tm_mon;          /* month */
    INT16          tm_year;         /* full year */
} Date_time_type;

#define Send_time_type      Date_time_type
#define IMMEDIATE          (-1)      /* Set into tm_sec of above struct */
#define URGENT             (-2)

/*
**      Error_type
**      The definition of the Error type does not include the text part of the error message; If the text
**      part is needed, it shall be treated as a private extension!
*/

typedef UINT32 Error_type;

```

```
/*
** La_id_type
** Reference of an LA-ID. Presented as a string.
** The purpose of the parameter is to identify the "owning" LA of a request.
*/
```

```
typedef CHAR La_id_type[15+1];
```

```
/*
** Layout_id_type
** Defines the layout of the <CopyTDD> target file.
*/
```

```
typedef enum { std=0, csv, tab } Layout_id_type;
```

```
/*
** Path_type
** Full path addressing a file of a directory. Stands for full path to document, file or directory.
** Full path means: path given absolute, without relative components.
*/
```

```
typedef CHAR Path_type[255+1];
```

```
/*
** Printer_id_type
** ID of selected printer. Represented as a string. Depends on the supporting operating system.
** The CA manufacturer shall state in documentation how to address printers.
*/
```

```
typedef CHAR Printer_id_type[127+1];
```

```
/*
** Req_id_type
** Reference of a Request ID. The parameter value is represented as a string encoded as implied
** by the APPLI/COM header ID of the TDD. The purpose of the parameter is to identify the
** relation of a response to a previous request. So the REQ-ID shall be unique within an LA. It is the
** responsibility of the LA to ensure the REQ-ID is unique.
*/
```

```
typedef CHAR_Req_id_type[31+1];
```

```
/*
** Service_type
** Specifies the ITU-T Service to be used.
*/
```

```
typedef enum { fx3=1, fx4, ttx, tlx, tx, email, ft } Service_type;
```

```
/*
** State_type
** Specifies the state of the CA-Record. See also 5.6.
*/
```

```
typedef enum { delayed=1, sending, sent, send_failed, reception, retrieved, receive_failed } State_type;
```

```
/*
** Status_type
*/
```

```
typedef enum { unknown, positive, partial, negative } Status_type;
```

```

/*
**      Subfunc_type
*/

```

```
typedef UINT32 Subfunc_type;
```

```

/*
**      Type_id_type
**      Specifies the Type (Subtype) of a telecommunications service. Each telecommunications
**      service has its own set of Type-ids. See appropriate clause of Part II of this Recommendation.
*/

```

```
typedef enum{ std=0, btm, dtm, bft, edi, opd, md, ctl, teletex, g3fax, g4class1, videotex, message, bilateral, national, oda
} Type_id_type;
```

```

/*
**      Userkey_type
*/

```

```
typedef CHAR_Userkey_type[31+1];
```

18.1.4 Structures TDD, constantes et définitions du service FX3

```

****
*      FX3: Service specific structure for <SendTDD>
****/

```

```

struct send_fx3_inp {
    G3_speed_type           g3speed;           /* <G3Speed> */
    Boolean_type            gencil;             /* <GenCil> */
    G3_high_res_type       highres;           /* <HighRes> */
    G3_sub_address_type    subaddr;           /* <SubAddress> */
    Boolean_type            useecm;           /* <UseEcm> */
    Address_type            address;           /* <Recipient> or <RecipientSpec> */
    Boolean_type            dopoll;           /* <DoPoll> */
    Poll_password_type     password;          /* <PollPassword> */
    Poll_select_type       select;           /* <PollSelector> */
    Path_type              filename;          /* <Document> or <DocumentSpec> */
    Convert_id_type        convert;          /* <Convert> */
    Type_id_type            type;             /* <Type> */
    INT16                  from;             /* <From> */
    INT16                  to;              /* <To> */
};

```

```

****
*      FX3: Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_fx3_inp {
    G3_speed_type           g3speed;           /* <G3Speed> */
    Boolean_type            gencil;             /* <GenCil> */
    G3_high_res_type       highres;           /* <HighRes> */
    G3_sub_address_type    subaddr;           /* <SubAddress> */
    Boolean_type            useecm;           /* <UseEcm> */
    Address_type            address;           /* <Recipient> */
    Boolean_type            dopoll;           /* <DoPoll> */
    Poll_password_type     password;          /* <PollPassword> */
    Poll_select_type       select;           /* <PollSelector> */
    Path_type              filename;          /* <Document> or <DocumentSpec> */
};

```

```

Convert_id_type      convert;      /* <Convert> */
Type_id_type        type;          /* <Type> */
INT16               from;         /* <From> */
INT16               to;          /* <To> */
};

struct sendack_fx3_out {
    Address_type      address;      /* <Recipient> */
    G3_speed_type    g3speed;      /* <G3Speed> */
    Boolean_type     useecm;       /* <UseEcm> */
};

/**
 *   FX3: Service specific structures for <ReceiveTDD>
 */

struct receive_fx3_inp {
    Convert_id_type   cvfax3;      /* <CvFax3> */
    G3_sub_address_type subaddr;   /* <SubAddress> */
    Path_type        filename;     /* <Document> */
};

struct receive_fx3_out {
    Address_type      address;      /* <Originator> */
    G3_speed_type    g3speed;      /* <G3Speed> */
    G3_sub_address_type subaddr;   /* <SubAddress> */
    Path_type        filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type   convert;     /* <Convert> */
    Type_id_type     type;         /* <Type> */
};

/**
 *   FX3: Structures for poll <ExtendTDD>
 */

struct poll_fx3_inp {
    Address_type      address;      /* <Address> */
    Poll_password_type password;    /* <PollPassword> */
    Poll_select_type  select;       /* <PollSelector> */
    Date_time_type    sendtime;     /* <SendTime> (optional) */
};

struct poll_fx3_out {
    Comid_type       comid;        /* <ComId> */
};

/**
 *   Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 *   Use of some definitions made in 18.1.3 are restricted. This applies to the following
 *   definitions:
 *
 *   Service_type is restricted to:      fx3.
 *   Type_id_type is restricted to:     std, btm, dtm, bft and edi.
 *   Convert_id_type:                   See Table 78 for further information.
 */

/*
**   G3_speed_type
*/

```

```
typedef enum {bps2400=1, bps4800, bps7200, bps9600, bps12200, bps14400} G3_speed_type;
```

```

/*
**      G3_sub_address_type
*/

```

```
typedef CHAR G3_sub_address_type[23+1];
```

```

/*
**      G3_high_res_type
*/

```

```
typedef enum { g3dpi98=0, g3dpi196, g3dpi200, g3dpi300, g3dpi400, g3dpi392_8, g3dpi392_16 } G3_high_res_type;
```

```

/*
**      Poll_password_type
*/

```

```
typedef CHAR Poll_password_type[23+1]; /* only first 20 digits are valid */
```

```

/*
**      Poll_select_type
*/

```

```
typedef CHAR_Poll_password_type[23+1]; /* only first 20 digits are valid */
```

18.1.5 Structures TDD, constantes et définitions du service FX4

```

****
*      FX4: Service specific structure for <SendTDD>
****/

```

```

struct send_fx4_inp {
    G4_high_res_type      highres;      /* <HighRes> */
    G4_sub_address_type   subaddr;      /* <SubAddress> */
    Address_type          address;      /* <Recipient> or <RecipientSpec> */
    Path_type             filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type       convert;      /* <Convert> */
    Type_id_type          type;         /* <Type> */
    String12_type         name;         /* <Name> */
    String12_type         userinfo;     /* <UserInfo> */
    Path_type             prolog;       /* <Prolog> */
    INT16                 from;         /* <From> */
    INT16                 to;          /* <To> */
};

```

```

****
*      FX4: Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_fx4_inp {
    G4_high_res_type      highres;      /* <HighRes> */
    G4_sub_address_type   subaddr;      /* <SubAddress> */
    Address_type          address;      /* <Recipient> */
    Path_type             filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type       convert;      /* <Convert> */
    Type_id_type          type;         /* <Type> */
    String12_type         name;         /* <Name> */
    String12_type         userinfo;     /* <UserInfo> */
};

```

```

    Path_type          prolog;          /* <Prolog> */
    INT16              from;           /* <From> */
    INT16              to;            /* <To> */
};

struct sendack_fx4_out {
    Cil_type           cil;           /* <Cil> */
};

/**
 *   FX4: Service specific structures for <ReceiveTDD>
 */

struct receive_fx4_inp {
    Convert_id_type    cvfax4;        /* <CvFax4> */
    G4_sub_address_type subaddr;      /* <SubAddress> */
    Path_type          filename;      /* <Document> */
    Path_type          prolog;        /* <Prolog> */
};

struct receive_fx4_out {
    Address_type       address;        /* <Originator> */
    G4_sub_address_type subaddr;      /* <SubAddress> */
    Path_type          filename;      /* <Document> or <DocumentSpec> */
    Convert_id_type    convert;       /* <Convert> */
    Type_id_type       type;          /* <Type> */
    String12_type      name;         /* <Name> */
    String12_type      userinfo;     /* <UserInfo> */
    Cil_type           cil;          /* <Cil> */
    INT16              firstpg;      /* <FirstPg> */
};

/**
 *   Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 *   Use of some definitions made in 18.1.3 are restricted. This applies to the following
 *   definitions:
 *
 *       Service_type is restricted to:      fx4.
 *       Type_id_type is restricted to:     std, dtm, bft, edi, opd, md, and ctl.
 *       Convert_id_type:                   See Table 87 for further information.
 */

/*
**   String12_type
*/

typedef CHAR String12_type[15+1];

/*
**   G4_sub_address_type
*/

typedef CHAR G4_sub_address_type[7+1];

/*
**   G4_high_res_type
*/

typedef enum {g4dpi200=1, g4dpi240, g4dpi300, g4dpi400} G4_high_res_type;

```

```

/*
**      Cil_type
*/

typedef struct {
    CHAR  receiver_tid[24];
    CHAR  sep1;           /* = '/' (slash) */
    CHAR  sender_tid[24];
    CHAR  sep2;           /* = '/' (slash) */
    CHAR  datetime[14];
    CHAR  sep3;           /* = '/' (slash) */
    CHAR  refinfo[7];
} Cil_type;           /* There is no terminating 0x00; Total length = 72 */

```

18.1.6 Structures TDD, constantes et définitions du service télétext

```

/**
 *      TTX: Service specific structure for <SendTDD>
 */

struct send_ttx_inp {
    Ttx_sub_address_type    subaddr;    /* <SubAddress> */
    Address_type            address;    /* <Recipient> or <RecipientSpec> */
    Path_type               filename;   /* <Document> or <DocumentSpec> */
    Convert_id_type         convert;    /* <Convert> */
    Type_id_type            type;       /* <Type> */
    String12_type           name;       /* <Name> */
    String12_type           userinfo;   /* <UserInfo> */
    Path_type               prolog;     /* <Prolog> */
    INT16                   from;       /* <From> */
    INT16                   to;        /* <To> */
    T61_options_type        t61options; /* <T61Options> */
};

/**
 *      TTX: Service specific structures for <SendAckTDD>
 */

struct sendack_ttx_inp {
    Ttx_sub_address_type    subaddr;    /* <SubAddress> */
    Address_type            address;    /* <Recipient> */
    Path_type               filename;   /* <Document> or <DocumentSpec> */
    Convert_id_type         convert;    /* <Convert> */
    Type_id_type            type;       /* <Type> */
    String12_type           name;       /* <Name> */
    String12_type           userinfo;   /* <UserInfo> */
    Path_type               prolog;     /* <Prolog> */
    INT16                   from;       /* <From> */
    INT16                   to;        /* <To> */
    T61_options_type};        t61options; /* <T61Options> */
};

struct sendack_ttx_out {
    Cil_type                cil;        /* <Cil> */
};

```

```

/**
 * TTX: Service specific structures for <ReceiveTDD>
 ***/

struct receive_ttx_inp {
    Convert_id_type    cvttx;        /* <CvTtx> */
    Ttx_sub_address_type subaddr;    /* <SubAddress> */
    Path_type          filename;    /* <Document> */
    Path_type          prolog;       /* <Prolog> */
};

struct receive_ttx_out {
    Address_type        address;     /* <Originator> */
    Ttx_sub_address_type subaddr;    /* <SubAddress> */
    Path_type          filename;    /* <Document> or <DocumentSpec> */
    Convert_id_type    convert;     /* <Convert> */
    Type_id_type        type;        /* <Type> */
    String12_type       name;        /* <Name> */
    String12_type       userinfo;    /* <UserInfo> */
    Cil_type            cil;         /* <Cil> */
    INT16               firstpg;     /* <FirstPg> */
};

/**
 * Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 * Use of some definitions made in 18.1.3 are restricted. This applies to the following
 * definitions:
 *
 * Service_type is restricted to:      ttx.
 * Type_id_type is restricted to:     std, dtm, bft and edi.
 * Convert_id_type:                   See Table 94 for further information.
 ***/

/*
 ** String12_type
 ** See definition of FX4!
 */
typedef CHAR String12_type[15+1];

/*
 ** Ttx_sub_address_type
 */
typedef CHAR Ttx_sub_address_type[7+1];

/*
 ** Cil_type
 ** See definition of FX4!
 */
typedef struct {
    CHAR receiver_tid[24];
    CHAR sep1          /* = '/' (slash) */
    CHAR sender_tid[24];
    CHAR sep2          /* = '/' (slash) */
    CHAR datetime[14];
    CHAR sep3          /* = '/' (slash) */
    CHAR refinfo[7];
} Cil_type;
*/

```

```

/*
**      T61_options_type
*/

```

```
typedef CHAR T61_options_type[15+1];
```

18.1.7 Structures TDD, constantes et définitions de l'identificateur des services télex obtenus par l'intermédiaire des services télételex (TX)

```

/**
*      TX: Service specific structure for <SendTDD>
***/

```

```

struct send_tx_inp {
    Boolean_type          notify;          /* <Notify> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Address_type         address;         /* <Recipient> or <RecipientSpec> */
    Path_type            filename;        /* <Document> or <DocumentSpec> */
    Convert_id_type      convert;         /* <Convert> */
    Type_id_type         type;            /* <Type> */
    INT16                from;           /* <From> */
    INT16                to;             /* <To> */
};

```

```

/**
*      TX: Service specific structures for <SendAckTDD>
***/

```

```

struct sendack_tx_inp {
    Boolean_type          notify;          /* <Notify> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Address_type         address;         /* <Recipient> */
    Path_type            filename;        /* <Document> or <DocumentSpec> */
    Convert_id_type      convert;         /* <Convert> */
    Type_id_type         type;            /* <Type> */
    INT16                from;           /* <From> */
    INT16                to;             /* <To> */
};

```

```
struct sendack_tx_out {};
```

```

/**
*      TX: Service specific structures for <ReceiveTDD>
***/

```

```

struct receive_tx_inp {
    Convert_id_type      cvtx;            /* <CvTx> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Path_type            filename;        /* <Document> */
};

```

```

struct receive_tx_out {
    Address_type         address;         /* <Originator> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Path_type            filename;        /* <Document> or <DocumentSpec> */
    Convert_id_type      convert;         /* <Convert> */
    Type_id_type         type;            /* <Type> */
};

```

```

/**
 * Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 * Use of some definitions made in 18.1.3 are restricted. This applies to the following
 * definitions:
 *
 *         Service_type is restricted to:      tx.
 *         Type_id_type is restricted to:     std
 *         Convert_id_type is restricted to:  "ASCII", "ASCIIxxx", "T.50"
 */

```

```

/*
** Tx_sub_address_type
*/

```

```
typedef CHAR Tx_sub_address_type[7+1];
```

18.1.8 Structures TDD, constantes et définitions du service télex

```

/**
 * TLX: Service specific structure for <SendTDD>
 */

```

```

struct send_tlx_inp {
    Tx_sub_address_type    subaddr;    /* <SubAddress> */
    Address_type           address;    /* <Recipient> or <RecipientSpec> */
    Path_type              filename;   /* <Document> or <DocumentSpec> */
    Convert_id_type        convert;    /* <Convert> */
    Type_id_type           type;       /* <Type> */
    INT16                  from;       /* <From> */
    INT16                  to;         /* <To> */
};

```

```

/**
 * TLX: Service specific structures for <SendAckTDD>
 */

```

```

struct sendack_tlx_inp {
    Tx_sub_address_type    subaddr;    /* <SubAddress> */
    Address_type           address;    /* <Recipient> */
    Path_type              filename;   /* <Document> or <DocumentSpec> */
    Convert_id_type        convert;    /* <Convert> */
    Type_id_type           type;       /* <Type> */
    INT16                  from;       /* <From> */
    INT16                  to;         /* <To> */
};

```

```
struct sendack_tlx_out {};
```

```

/**
 * TLX: Service specific structures for <ReceiveTDD>
 */

```

```

struct receive_tlx_inp {
    Convert_id_type        cvtlx;      /* <CvTx> */
    Tx_sub_address_type    subaddr;    /* <SubAddress> */
    Path_type              filename;   /* <Document> */
};

```

```

struct receive_tlx_out {
    Address_type          address;          /* <Originator> */
    Tlx_sub_address_type subaddr;          /* <SubAddress> */
    Path_type            filename;         /* <Document> or <DocumentSpec> */
    Convert_id_type      convert;          /* <Convert> */
    Type_id_type         type;             /* <Type> */
};

```

/**

* **Constants to be used for specific fields of the Send, Sendack and Receive TDD.**
 * Use of some definitions made in 18.1.3 are restricted. This applies to the following
 * definitions:

```

*           Service_type is restricted to:      tlx.
*           Type_id_type is restricted to:     std
*           Convert_id_type is restricted to:  "ASCII", "ASCIIxxx", "T.50"
***/

```

/*

** **Tlx_sub_address_type**

*/

```

typedef CHAR Tlx_sub_address_type[7+1];

```

18.1.9 Structures TDD, constantes et définitions du service EMAIL

/**

* EMAIL: Service specific structure for <SendTDD>

***/

```

struct send_email_inp {
    Ipm_id_type          ipm_id;           /* <IpmId> */
    Path_type           s_recipient;      /* <S-Recipient> */
    Boolean_type        alternate;        /* <Alternate> (optional) */
    Cont_type_type     cont_type;         /* <ContType> (optional) */
    Boolean_type        disclo_rec;       /* <DiscloRec> (optional) */
    Date_time_type     expirytime;       /* <ExpiryTime> (optional) */
    Boolean_type        implicit_conv;    /* <ImplicitConv> (optional) */
    Importance_type    importance;        /* <Importance> (optional) */
    Language_id_type   language;         /* <Language> (optional) */
    Priority_type       priority;         /* <Priority> (optional) */
    Path_type          related;          /* <Related> (optional) */
    Ipm_id_type        reply_id;         /* <ReplyId> (optional) */
    Date_time_type     replytime;        /* <ReplyTime> (optional) */
    Path_type          s_originator;     /* <S-Originator> (optional) */
    Sensitivity_type   sensitivity;      /* <Sensitivity> (optional) */
    Subject_type       subject;          /* <Subject> (optional) */
    Userinfo_type      userinfo;         /* <UserInfo> (optional) */
};

```

/**

* EMAIL:Service specific structures for <SendAckTDD>

***/

```

struct sendack_email_inp {
    Ipm_id_type          ipm_id;           /* <IpmId> */
    Path_type           s_recipient;      /* <S-Recipient> */
    Boolean_type        alternate;        /* <Alternate> (optional) */
    Cont_type_type     cont_type;         /* <ContType> (optional) */
};

```

```

Boolean_type      disclo_rec;      /* <DiscloRec> (optional) */
Date_time_type    expirytime;     /* <ExpiryTime> (optional) */
Boolean_type      implicit_conv;  /* <ImplicitConv> (optional) */
Importance_type   importance;     /* <Importance> (optional) */
Language_id_type  language;       /* <Language> (optional) */
Priority_type      priority;       /* <Priority> (optional) */
Path_type         related;        /* <Related> (optional) */
Ipm_id_type       reply_id;       /* <ReplyId> (optional) */
Date_time_type    replytime;      /* <ReplyTime> (optional) */
Path_type         s_originator;    /* <S-Originator> (optional) */
Sensitivity_type  sensitivity;    /* <Sensitivity> (optional) */
Subject_type      subject;        /* <Subject> (optional) */
Userinfo_type     userinfo;       /* <UserInfo> (optional) */
};

```

```

struct sendack_email_out {
    Msg_sub_id_type    msg_sub_id; /* <MsgSubId> (optional) */
    Date_time_type     submittime; /* <SubmitTime> (optional) */
};

```

```

/**
 * EMAIL:Service specific structures for <ReceiveTDD>
 */

```

```

struct receive_email_inp {
    Path_type          r_originator; /* <R-Originator> */
    Path_type          r_recipient; /* <R-Recipient> */
    Path_type          related;     /* <Related> (optional) */
};

```

```

struct receive_email_out {
    Ipm_id_type        ipm_id;      /* <IpmId> */
    Cont_type_type     cont_type;   /* <ContType> */
    Date_time_type     expirytime; /* <ExpiryTime> */
    Boolean_type        forwarded; /* <Forwarded> */
    Importance_type     importance; /* <Importance> */
    Boolean_type        inc_copy;   /* <IncCopy> */
    Language_id_type   language;   /* <Language> */
    Priority_type       priority;   /* <Priority> */
    Ipm_id_type        reply_id;   /* <ReplyId> */
    Date_time_type     replytime;  /* <ReplyTime> */
    Sensitivity_type    sensitivity; /* <Sensitivity> */
    Subject_type        subject;    /* <Subject> */
    Date_time_type     submittime; /* <SubmitTime> */
    Userinfo_type       userinfo;   /* <UserInfo> */
};

```

```

/**
 * EMAIL: Structure of file-record of R-Recipientspec
 */

```

```

struct r_recipientspec {
    X_name_type        x_name;
    R_recipient_type   r_type;
    Boolean_type        reply;
    Notify_type        notify;
    Report_type        report;
};

```

```
/**
 * EMAIL: Structure of file-record of S-Recipientspec
 ***/
```

```
struct s_recipientspec {
    X_name_type          x_name;
    S_recipient_type     s_type;
    Boolean_type         reply;
    Notify_type          notify;
    Report_type          report;
};
```

```
/**
 * EMAIL: Structure of file-record of Originatorspec
 ***/
```

```
struct originatorspec {
    X_name_type          x_name;
    Originator_type     o_type;
};
```

```
/**
 * EMAIL: Structure of file-record of Relatedspec
 ***/
```

```
struct relatedspec {
    Ipm_id_type         ipm_id;
    Relation_type       relation;
};
```

```
/**
 * Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 * Use of some definitions made in 18.1.3 are restricted. This applies to the following
 * definitions:
```

```
 *           Service_type is restricted to:      email.
 *           Type_id_type is restricted to:      std, teletex, g3fax, g4class1, videotex, message,
 *                                               bilateral, national
 *           Convert_id_type:                   See Table 113 for further information.
```

```
 ***/
```

```
/*
 ** Cont_type_type
 */
```

```
typedef enum { mhsIPM84=0, mhsIPM88, mhsIPN84, mhsIPN88, mhsDR, cmcIPM, cmcIPN, cmcDR }
    Cont_type_type;
```

```
/*
 ** Importance_type
 */
```

```
typedef enum { low = 0, normal, high } Importance_type;
```

```
/*
 ** Ipm_id_type
 */
```

```
typedef struct {
    BYTE id[63+1];
    BYTE oraddress[511+1];
} Ipm_id_type;
```

```

/*
**      Language_id_type
*/

typedef CHAR   Language_id_type[5+1];

/*
**      Msg_sub_id_type
*/

typedef CHAR   Msg_sub_id_type[67+1];

/*
**      Priority_type
*/

typedef enum { standard=0, nonurgent, urgent } Priority_type;

/*
**      Relation_type
*/

typedef enum { reference=0, obsolete } Relation_type;

/*
**      Subject_type
*/

typedef CHAR   Subject_type[127+1];

/*
**      Sensitivity_type
*/

typedef enum { none=0, personal, privateonly, companyconfidential } Sensitivity_type;

/*
**      Userinfo_type
*/

typedef CHAR   Userinfo_type[15+1];

/*
**      X_name_type
*/

typedef struct {
    Boolean_type          is_alias;
    CHAR                  addr_or_alias[511+1];
} X_name_type;

/*
**      S_recipient_type
*/

typedef enum { sndprimary=1, sndcopy, sndblind } S_recipient_type;

```

```

/*
**      R_recipient_type
*/

```

```
typedef enum { recprimary=1, reccopy, recblind, recintended } R_recipient_type;
```

```

/*
**      Notify_type
*/

```

```
typedef enum { nonotify=1, notreceived, received, explicit } Notify_type;
```

```

/*
**      Report_type
*/

```

```
typedef enum { noreport=1, basic, confirmed } Report_type;
```

```

/*
**      Originator_type
*/

```

```
typedef enum { authorizing=1, originator, replyto } Originator_type;
```

18.1.10 Structures TDD, constantes et définitions de l'identificateur des services de transfert de fichiers (FT)

```

****
*      FT: Service specific structure for <SendTDD>
****/

```

```

struct send_ft_inp {
    Environ_type          environ;          /* <Environ> */
    Password_type        password;         /* <Password> (optional) */
    Address_type         address;          /* <Recipient> or <RecipientSpec> */
    Path_type            filename;         /* <Document> or <DocumentSpec> */
    Convert_id_type      convert;          /* <Convert> */
    Type_id_type         type;             /* <Type> */
    Compress_type        compress;         /* <Compress> */
    Name_type            name;             /* <Name> (optional) */
};

```

```

****
*      FT: Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_ft_inp {
    Environ_type          environ;          /* <Environ> */
    Password_type        password;         /* <Password> (optional) */
    Address_type         address;          /* <Recipient> */
    Path_type            filename;         /* <Document> or <DocumentSpec> */
    Convert_id_type      convert;          /* <Convert> */
    Type_id_type         type;             /* <Type> */
    Compress_type        compress;         /* <Compress> */
    Name_type            name;             /* <Name> (optional) */
};

```

```
struct sendack_ft_out { };
```

```

/**
 * FT: Service specific structures for <ReceiveTDD>
 ***/

struct receive_ft_inp {
    Password_type      password;      /* <Password> (optional) */
    Path_type          filename;      /* <Document> (optional) */
};

struct receive_ft_out {
    Environ_type       environ;       /* <Environ> */
    Address_type       address;       /* <Originator> */
    Path_type          filename;      /* <Document> or <DocumentSpec> */
    Convert_id_type    convert;       /* <Convert> */
    Type_id_type       type;          /* <Type> */
    Compress_type      compress;      /* <Compress> */
    Name_type          name;          /* <Name> */
};

/**
 * Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 * Use of some definitions made in 18.1.3 are restricted. This applies to the following
 * definitions:
 *
 * Service_type is restricted to:      ft
 * Type_id_type is restricted to:     std
 * Convert_id_type:                   std, void.
 ***/

/*
 ** Compress_type
 * See also 17.2.2.6
 */

typedef CHAR Compress_type[7+1];

/*
 ** Environ_type
 */

typedef enum { msdos=1, windows, unix, os2, macos } Environ_type;

/*
 ** Password_type
 */

typedef CHAR Password_type[15+1];

/*
 ** Name_type
 */

typedef CHAR Name_type[15+1];

```

PARTIE IV – PLATFORM DEPENDENCIES (DÉPENDANCE DES PLATES-FORMES)

19 Facteurs de dépendance à l'égard de la mise en œuvre

Le présent article décrit les facteurs de dépendance des différentes plates-formes à l'égard de la mise en œuvre.

La présente Recommandation n'évoque la question de la dépendance des plates-formes que dans le point qu'elle consacre à la mise en œuvre de la méthode d'échange de primitives (voir 7.3.1). La méthode d'échange de primitives permet l'accès direct aux fonctions de la méthode d'échange de base décrites au 7.1. Le paragraphe 19.3 décrit la mise en œuvre de la méthode d'échange de primitives pour chaque plate-forme.

La mise en œuvre de la méthode d'échange de primitives est facultative pour certaines plates-formes, car la méthode d'échange de fichiers qui offre une plus grande indépendance des plates-formes peut avoir été choisie comme méthode d'échange par défaut pour certaines de ces plates-formes. Les méthodes d'échange par défaut assignées aux différentes plates-formes sont indiquées au 19.2.

Le système de codage binaire décrit dans la Partie III de la présente Recommandation ne dépend pas de la plate-forme. Toutefois, étant donné qu'on a utilisé un langage C générique pour la description de ce codage, nous commencerons par indiquer au 19.1 ci-dessous la correspondance entre les types de données C génériques et les types de données réels.

19.1 Mise en correspondance des types de données des descriptions TDD à codage binaire

Voir le Tableau 125.

TABLEAU 125/T.611

Mise en correspondance des types de données C génériques pour les différentes plates-formes

Plate-forme	Type de données C réel correspondant au type de données C générique			
	INT16	UINT32	CHAR	BYTE
MS-DOS	short int	unsigned long int	char	unsigned char
WINDOWS	short int	dword	char	byte
Unix	short int	unsigned long int	char	unsigned char
OS/2	short int	unsigned long int	char	unsigned char
MacOS	Appelle un complément d'étude			

19.2 Méthode d'échange par défaut

Voir le Tableau 126.

TABLEAU 126/T.611

Méthodes d'échange par défaut assignées aux différentes plates-formes

Plate-forme	Méthode d'échange par défaut
MS-DOS	Méthode d'échange de fichiers
WINDOWS	Méthode d'échange de primitives
Unix	Appelle un complément d'étude
OS/2	Appelle un complément d'étude
MacOS	Appelle un complément d'étude

19.3 Mise en œuvre de la méthode d'échange de primitives

On mettra en œuvre la méthode d'échange de primitives en donnant accès aux fonctions de la méthode d'échange de base qui sont décrites de manière générale au 7.1. Pour assurer la compatibilité binaire à travers une plate-forme donnée, il convient de respecter strictement les règles, les types de données et les prototypes de fonctions spécifiés dans les paragraphes qui suivent pour chaque plate-forme.

Pour fonctionner convenablement, la mise en œuvre doit s'appuyer sur certaines structures de données et certaines constantes. Les structures de données et les constantes communes à toutes les mises en œuvre sont indiquées ci-dessous dans la description en langage C:

```
/*
** This generic C language description uses certain generic types which are mapped to
** real data types in the platform dependent descriptions further below.
** The generic types used are: NUMBER, BUFPOINTER, LISTPOINTER and WORD.
*/

/**
* The structure bem_data_buffer describes a buffer.
* The types NUMBER and BUFPOINTER have to be mapped to each platform
***/

struct bem_data_buffer {
    NUMBER          buffer_size;
    BUFPOINTER      buffer;
};

/**
* The structure bem_data_files describes a list of filenames.
* The types NUMBER and LISTPOINTER have to be mapped to each platform
***/

struct bem_data_files {
    NUMBER          number_of_files;
    LISTPOINTER     file_name[ ];
};

/**
* The structure bem_data_descriptor describes the data conveyed between LAs and CAs
* (data files, address lists, etc.). The data may be in files and/or in memory. The structure
* bem_data_descriptor accounts for these situations.
*
* When data are in memory, the structure bem_data_descriptor allows to actually convey that
* data through the EPutData or EGetData functions. When data are in files, bem_data_descriptor
* describes the actual file names that contain that data, i.e., the structure does not contain the data
* itself.
*
* The type WORD denotes a 16 bit unsigned integer, which has to be mapped to each platform
***/

struct bem_data_descriptor {
    WORD            key;          /* key = 0 for data in buffer */
                                /* key = 1 for data in file */
    WORD            type;        /* type = 1 for "document lists" */
                                /* type = 2 for "recipient lists", etc. */
                                /* used only in SEND, SENDACK and RECEIVE */
                                /* ignored otherwise */
    union {
        struct bem_data_buffer buffer; /* data in memory buffer */
        struct bem_data_files file;    /* data in file */
    } bof; /* buffer or file */
};
```

```

#define BEM_DATA      struct bem_data_descriptor

/* Constants used for the "key" field of BEM_DATA */
#define BUFFER_KEY    0      /* key: data in buffer */
#define FILE_KEY      1      /* key: data in file */

/* Constants used for the "type" field of BEM_DATA */
#define DOCUMENT_TYPE 1      /* type: document list */
#define RECIPIENT_TYPE 2     /* type: recipient list (= address list) */
#define ORIGINATOR_TYPE 3    /* type: originator list */
#define RELATED_TYPE  4      /* type: related list */

/* Constants used for TDD Types, see also 7.2.4.2 */
#define NO_TDD_RESPONSE 0x00
#define SENDACK_RESPONSE 0x10
#define RECEIVE_RESPONSE 0x20
#define COPY_RESPONSE 0x30
#define DELETE_RESPONSE 0x31
#define CANCEL_RESPONSE 0x32
#define PURGE_RESPONSE 0x33
#define RESCHEDULE_RESPONSE 0x34
#define DISPATCH_RESPONSE 0x35
#define PREVIEW_RESPONSE 0x36
#define PRINT_RESPONSE 0x40
#define CONVERT_RESPONSE 0x41
#define CHECK_RESPONSE 0x42
#define EXTEND_RESPONSE 0x50
#define NATIONAL_RESPONSE 0x60
#define PRIVATE_RESPONSE 0x70

/* Constants used for Alarm_types (may be or'ed), see also 7.2.7.1 */
#define ASYNC_RESPONSES 0x0001
#define QUEUE_FULL 0x0002
#define DOCUMENT_RECEIVED 0x0004
#define CONNECTION_LOST 0x0008
#define SEND_SUCCESS 0x0010
#define SEND_FAILED 0x0020
#define CORRUPTED_TDD 0x0040
#define SEND_EVENT 0x0080
#define RECEIVE_EVENT 0x0100
#define CA_WILL_STOP 0x0200
#define ALARMS_UNAVAILABLE 0x0400
#define TDD_RESP_AVAILABLE 0x0800

```

NOTE – Traitement du paramètre Data-ID dans la fonction EPutTDD à l'entrée et à la sortie. A l'entrée: l'application locale (LA) définit une adresse non-NULL si cette LA a besoin de transférer des fichiers de données dans des appels de fonctions EPutData ultérieurs. Si l'application locale n'a pas besoin de transférer des fichiers, ce paramètre est sur NULL. A la sortie: si le paramètre d'entrée était sur NULL, l'application de communication (CA) ne modifie pas cette valeur. Sinon, la CA calcule un paramètre Data-ID et le renvoie à l'adresse spécifiée dans le paramètre d'entrée.

19.3.1 MSDOS

L'accès à la méthode d'échange de base est assuré par l'intermédiaire d'un mécanisme d'interruption DOS. L'application locale appelante doit respecter les règles suivantes:

- le numéro d'interruption utilisé par l'application de communication doit être extrait de l'ICE (descripteur de CA);
- en cas d'utilisation d'un système de multiplexage, le numéro de multiplexage doit être extrait de l'ICE (descripteur de CA) et chargé dans l'enregistreur AH avant appel;

- pour être transférés à l'application de communication, les paramètres doivent être placés sur la pile de l'abonné appelant. Cette pile doit pouvoir recevoir encore au moins 256 multiplats en cours d'appel;
- on doit utiliser les conventions d'appel C pour placer les paramètres (de droite à gauche); les pointeurs doivent être placés sous la forme d'adresses à deux mots (pointeurs d'extrémité);
- l'adressage d'une fonction voulue doit être assuré par l'intermédiaire d'un code de fonction, lui aussi placé sur la pile de l'abonné appelant en tant que paramètre de gauche;
- la valeur de renvoi doit être transmise dans l'enregistreur AX;
- les enregistreurs de segment (ainsi que le pointeur de pile) doivent être préservés par l'application de communication en cours d'appel; tous les autres enregistreurs peuvent être détruits par l'application de communication;
- le nettoyage de la pile (élimination des paramètres d'appel) doit être assuré par l'abonné appelant après le renvoi de la fonction (convention d'appel en langage C).

Pour accéder aux diverses fonctions, il convient d'utiliser les codes de fonction définis dans le Tableau 127.

TABLEAU 127/T.611
Codes de fonction DOS

Fonction de la méthode d'échange de base	Code de fonction
ELogin ()	1
EPutTDD ()	2
EPutData ()	3
EPollTDD ()	4
EGetTDD ()	5
EGetData ()	6
ESetAlarm ()	7
EAbortData ()	8
ELogout ()	9

19.3.1.1 Prototypes et définitions de la fonction C

```
#define STATUS      short int
```

```
#define BEM_DATA   struct bem_data_descriptor
```

```
struct bem_data_buffer {
    unsigned long      buffer_size;      /* = generic type NUMBER */
    unsigned char far * buffer;         /* = generic type BUFPOINTER */
};
```

```
struct bem_data_files {
    unsigned long      number_of_files; /* = generic type NUMBER */
    char far *        file_name[ ];    /* = generic type LISTPOINTER */
};
```

```

struct bem_data_descriptor {
    unsigned short int      key;                /* = generic type WORD */
    unsigned short int      type;              /* = generic type WORD */
    union {
        struct bem_data_buffer buffer;
        struct bem_data_files file;
    } bof;
};

```

***/

* ELogin ()

*

* Stack: SP+18	Connection_ID	far pointer
* SP+14	CA_ID	far pointer
* SP+10	Selector	far pointer
* SP+6	Password	far pointer
* SP+2	Login_name	far pointer
* SP -->	0x0001	word value; function code to identify ELogin ()

***/

```

STATUS ELogin (
    short int      function_code,          /* ELogin = 0x0001 */
    char far *     Login_name,
    char far *     Password,
    char far *     Selector,
    short int far * CA_ID,
    short int far * Connection_ID
);

```

***/

* EPutTDD ()

*

* Stack: SP+12	Data_ID	far pointer
* SP+10	TDD_size	word value
* SP+6	TDD_location	far pointer
* SP+4	CA_ID	word value
* SP+2	Connection_ID	word value
* SP -->	0x0002	word value; function code to identify EPutTDD ()

*

* If the address of Data_ID is set to NULL on call, CA will not provide the Data_ID and process the TDD immediately, assuming the pathes given in the TDD are correct

***/

```

STATUS EPutTDD (
    short int      function_code,          /* EPutTDD = 0x0002 */
    short int      Connection_ID,
    short int      CA_ID,
    unsigned char far * TDD_location,
    short int      TDD_size,
    unsigned long far * Data_ID
);

```

***/
*/

* EPutData ()

*/

* Stack:	SP+14	Next	word value
*	SP+10	Data	far pointer
*	SP+6	Data_ID	dword value
*	SP+4	CA_ID	word value
*	SP+2	Connection_ID	word value
*	SP -->	0x0003	word value; function code to identify EPutData ()

***/
*/

STATUS EPutData (

short int	function_code,	/* EPutData = 0x0003 */
short int	Connection_ID,	
short int	CA_ID,	
unsigned long	Data_ID,	
BEM_DATA far *	Data,	
short int	Next	

);

***/
*/

* EPolITDD ()

*/

* Stack:	SP+14	TDD_count	far pointer
*	SP+10	TDD_size	far pointer
*	SP+6	TDD_type	far pointer
*	SP+4	CA_ID	word value
*	SP+2	Connection_ID	word value
*	SP -->	0x0004	word value; function code to identify EPolITDD ()

***/
*/

STATUS EPolITDD (

short int	function_code,	/* EPolITDD = 0x0004 */
short int	Connection_ID,	
short int	CA_ID,	
unsigned short far *	TDD_type,	
short int far *	TDD_size,	
short int far *	TDD_count	

);

***/
*/

* EGetTDD ()

*/

* Stack:	SP+14	Data_ID	far pointer
*	SP+10	TDD_size	far pointer
*	SP+6	TDD_location	far pointer
*	SP+4	CA_ID	word value
*	SP+2	Connection_ID	word value
*	SP -->	0x0005	word value; function code to identify EGetTDD ()

***/
*/

```

STATUS      EGetTDD (
short int    function_code,          /* EGetTDD = 0x0005 */
short int    Connection_ID,
short int    CA_ID,
unsigned char far * TDD_location,
short int far * TDD_size,
unsigned long far * Data_ID
);

```

***/

```

*      EGetData ()

```

*

```

*      Stack:  SP+14      Next          far pointer
*                  SP+10      Data          far pointer
*                  SP+6       Data_ID       dword value
*                  SP+4       CA_ID         word value
*                  SP+2       Connection_ID word value
*                  SP -->     0x0006      word value; function code to identify EGetData ()

```

***/

```

STATUS      EGetData (
short int    function_code,          /* EGetData = 0x0006 */
short int    Connection_ID,
short int    CA_ID,
unsigned long Data_ID,
BEM_DATA far * Data,
short int far * Next
);

```

***/

```

*      ESetAlarm ()

```

*

```

*      Stack:  SP+8      Alarm_handler  far pointer
*                  SP+6      Alarm_event  word value
*                  SP+4      CA_ID         word value
*                  SP+2      Connection_ID word value
*                  SP -->     0x0007      word value; function code to identify ESetAlarm ()

```

***/

```

void        FAR AlarmHandler (
short int    Connection_ID,
short int    CA_ID,
unsigned short Alarm_type,
unsigned char far * Parameter,      /* additional parameter buffer */
short int    Length                /* length of parameter buffer */
);

```

```

STATUS      ESetAlarm (
short int    function_code,          /* ESetAlarm = 0x0007 */
short int    Connection_ID,
short int    CA_ID,
unsigned short Alarm_event,
void (* far Alarm_handler) ()
);

```

```

***/
*      EAbortData ( )
*
*      Stack:  SP+6      

|               |
|---------------|
| Data_ID       |
| CA_ID         |
| Connection_ID |
| 0x0008        |

      dword value
*
*              SP+4      

|       |
|-------|
| CA_ID |
|-------|

      word value
*
*              SP+2      

|               |
|---------------|
| Connection_ID |
|---------------|

      word value
*
*              SP -->    

|        |
|--------|
| 0x0008 |
|--------|

      word value; function code to identify EAbortData ( )
***/

```

```

STATUS      EAbortData (
short int      function_code,      /* EAbortData = 0x0008 */
short int      Connection_ID,
short int      CA_ID,
unsigned long  Data_ID
);

```

```

***/
*      ELogout ( )
*
*      Stack:  SP+4      

|               |
|---------------|
| CA_ID         |
| Connection_ID |
| 0x0009        |

      word value
*
*              SP+2      

|               |
|---------------|
| Connection_ID |
|---------------|

      word value
*
*              SP -->    

|        |
|--------|
| 0x0009 |
|--------|

      word value; function code to identify ELogout ( )
***/

```

```

STATUS      ELogout (
short int      function_code,      /* ELogout = 0x0009 */
short int      Connection_ID,
short int      CA_ID
);

```

19.3.2 WINDOWS

Pour accéder aux fonctions de la méthode d'échange de base, le fournisseur de l'application de communication doit offrir une bibliothèque à liaison dynamique (DLL) (*dynamic link library*), qui s'oppose aux fonctions de la méthode d'échange avec l'application locale décrites dans le paragraphe suivant.

La personne qui a décidé de la configuration de la plate-forme cible doit indiquer le nom de la bibliothèque DLL à l'emplacement approprié dans l'ICE. Pour pouvoir accéder aux fonctions DLL, l'application locale doit tout d'abord émettre un appel de fonction du système Windows "LoadLibrary", en utilisant comme paramètre le nom de la bibliothèque DLL choisie.

19.3.2.1 Prototypes et définitions de la fonction C

```

#include <windows.h>

#define BEM_DATA struct bem_data_descriptor

struct bem_data_buffer {
int      buffer_size;      /* = generic type NUMBER */
HANDLE   buffer;          /* = generic type BUFPOINTER */
};

struct bem_data_files {
int      number_of_files; /* = generic type NUMBER */
LPSTR    file_name[1];   /* = generic type LISTPOINTER */
};

```

```

struct bem_data_descriptor {
    word                key;                /* = generic type WORD */
    word                type;              /* = generic type WORD */
    union {
        struct bem_data_buffer buffer;
        struct bem_data_files file;
    } bof;
};

/**
 *      ELogin ( )
 ***/

void FAR PASCAL ELogin (
    LPSTR                Login_name,        /* User name */
    LPSTR                Password,         /* User's password */
    LPSTR                Selector,         /* CA Selector */
    int far *            CA_ID,           /* CA Identifier */
    int far *            Connection_ID,    /* Connection ID */
    int far *            Status           /* 0 success, 1 fail */
);

/**
 *      EPutTDD ( )
 *
 *      If the address of Data_ID is set to NULL on call, CA will not provide the Data_ID and process
 *      the TDD immediately, assuming the pathes given in the TDD are correct
 ***/

void FAR PASCAL EPutTDD (
    int                Connection_ID,      /* Connection ID */
    int                CA_ID,             /* CA Identifier */
    LPSTR              TDD_location,      /* TDD buffer text */
    int                TDD_size,          /* size of the TDD text */
    int far *          Data_ID,           /* Data identifier */
    int far *          Status             /* 0 success, 1 fail */
);

/**
 *      EPutData ( )
 ***/

void FAR PASCAL EPutData (
    int                Connection_ID,      /* Connection ID */
    int                CA_ID,             /* CA Identifier */
    int                Data_ID,           /* Data identifier */
    BEM_DATA far *    Data,               /* Data descriptor */
    int                Next,              /* next boolean */
    int far *          Status             /* 0 success, 1 fail */
);

/**
 *      EPollTDD ( )
 ***/

```

```

void FAR PASCAL EPollTDD (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    word far *   TDD_type,     /* Type of next TDD */
    int far *   TDD_size,     /* Size of next TDD */
    int far *   TDD_count,    /* Count of waiting TDDs */
    int far *   Status        /* 0 success, 1 fail */
);

/**
 *      EGetTDD ( )
 ***/

void FAR PASCAL EGetTDD (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    LPSTR       TDD_location, /* TDD buffer text */
    int far *   TDD_size,     /* size of the TDD text */
    int far *   Data_ID,     /* Data identifier */
    int far *   Status        /* 0 success, 1 fail */
);

/**
 *      EGetData ( )
 ***/

void FAR PASCAL EGetData (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    int          Data_ID,     /* Data identifier */
    BEM_DATA far * Data,     /* Data descriptor */
    int far *   Next,        /* next boolean */
    int far *   Status        /* 0 success, 1 fail */
);

/**
 *      ESetAlarm ( )
 ***/

void FAR PASCAL AlarmHandler (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    word        Alarm_type,    /* Alarm type */
    HANDLE      Parameter,    /* additional parameter buffer */
    int         Length        /* length of parameter buffer */
);

void FAR PASCAL ESetAlarm (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    word        Alarm_type,    /* Alarm event */
    FARPROC     Alarm_handler, /* Alarm handler entry */
    int far *   Status        /* 0 success, 1 fail */
);

/**

```

```

*      EAbortData ( )
***/

void FAR PASCAL EAbortData (
    int      Connection_ID,      /* Connection ID */
    int      CA_ID,              /* CA Identifier */
    int      Data_ID,           /* Data identifier */
    int far * Status             /* 0 success, 1 fail */
);
/****

*      ELogout ( )
***/

void FAR PASCAL ELogout (
    int      Connection_ID,      /* Connection ID */
    int      CA_ID,              /* CA Identifier */
    int far * Status             /* 0 success, 1 fail */
);

```

19.3.3 UNIX

Appelle un complément d'étude.

19.3.4 OS/2

Appelle un complément d'étude.

19.3.5 MacOS

Appelle un complément d'étude.

Annexe A

Syntaxe pour la présentation et le codage

(Cette annexe fait partie intégrante de la présente Recommandation)

A.1 Syntaxe à forme de Backus-Naur (BNF)

Afin de fournir une description générique, on fait appel à une syntaxe à forme de Backus-Naur. Les règles générales de la forme de Backus-Naur, décrites ci-dessous, sont employées dans l'ensemble de la présente Recommandation.

- Un jeton terminal (feuille) est noté par un littéral. Si celui-ci ne peut pas être distingué des littéraux utilisés pour décrire la syntaxe, le jeton doit être délimité par des apostrophes doubles ("").
- Des chaînes de caractères délimitées par des apostrophes doubles désignent des jetons de terminal se composant du texte constant de ces chaînes.
- Un jeton non terminal (nœud) est noté par un littéral terminé par les caractères "<" (inférieur à) et ">" (supérieur à).
- Un jeton (ou groupe de jetons) facultatif est délimité par les caractères "[" et "]".
- Un groupe de jetons est délimité par les caractères "(" et ")". Les groupes peuvent être imbriqués.
- Un groupe de jetons délimités par "{" et "}" peut être répété 0, 1 ou plusieurs fois.
- Le caractère "|" est utilisé pour séparer des jetons (ou groupes de jetons) en variante.

- La chaîne "[:=" est utilisée comme délimiteur de production.
- Les jetons sont séparés par des caractères d'espace ou de tabulation ou par une combinaison de ces deux types de caractères.
- La chaîne "--" est utilisée pour introduire un commentaire dans la description en BNF. Le commentaire se termine à la fin de la ligne. Les commentaires du BNF ne peuvent pas être imbriqués.

Concernant le codage en mode texte:

- Le jeton terminal STRING indique une chaîne de caractères codée comme l'implique la valeur de l'identificateur Code-ID (voir le Tableau 14). Cette chaîne de caractères peut avoir une longueur limitée, telle que l'indique le jeton STRING (SIZE (xxx.yyy)). Dans ce cas, xxx indique la longueur minimale de la chaîne, yyy indique la longueur maximale de la chaîne.
- La notation spéciale "STRING (xxxx + yyyy)", où "xxxx" et "yyyy" sont des chaînes de caractères, donne lieu à une chaîne formée par la concaténation de "xxxx" et "yyyy".
- Le jeton terminal PATH indique une chaîne (STRING) ne contenant que des caractères valides pour spécifier le chemin absolu pour le système d'exploitation sous-jacent.
- Le jeton terminal NUMERIC-STRING indique une chaîne de caractères représentant un nombre. Seuls les chiffres de 0 à 9 peuvent être utilisés.
- Le jeton ↵ indique une coupure de ligne. Il est toutefois possible d'inclure un commentaire dans des données de description de tâches (TDD) en mettant un caractère point-virgule à la fin de la ligne de description TDD, du commentaire proprement dit et de la coupure de ligne.
- L'ordre impliqué par les règles de production dans la description en BNF peut ne pas être l'ordre dans lequel les jetons devront être codés. Les règles additionnelles pour les schémas de codage spécifieront l'ordre éventuel qu'il faudra respecter.

A.2 Notation en langage C

La notation générique en langage C est fondée sur la norme ANSI actuelle. Si bien que les descriptions pourront être, une fois rassemblées dans un fichier, compilées par les compilateurs ANSI existants. Dans les sections dépendantes du système d'exploitation, il faudra toutefois appliquer les compilateurs appropriés. En général, les descriptions sont fondées sur les conventions suivantes:

- les 16 premiers caractères des définitions et des noms de variable sont significatifs;
- les majuscules et les minuscules sont différenciées.

Annexe B

Emplacement de l'environnement de configuration d'interface (ICE)

(Cette annexe fait partie intégrante de la présente Recommandation)

L'environnement de configuration d'interface (ICE) (*interface configuration environment*) représente une source "globale" pour toutes les applications locales (LA) conformes à la présente Recommandation.

Avec les systèmes d'exploitation UNIX, MacOS et les systèmes d'exploitation relevant de la famille Microsoft (OS/2, MS-DOS, etc.), l'ICE maître est représenté par un fichier¹⁷⁾. Dans un environnement de calculateurs en réseau de zone local (RZL) tel que NetwareTM, Vines ou LAN Manager, l'ICE maître est également présenté sous forme de fichier. Celui-ci peut être ouvert et lu par l'application afin d'en extraire les informations sur les applications de communication accessibles dans le système. Le nom de ce fichier, dans les systèmes ou réseaux ci-dessus mentionnés, est "APPLICOM.ICE".

¹⁷⁾ Dans ce contexte, un "fichier" est soit un fichier du système d'exploitation proprement dit ou un pilote de périphérique, qui se comporte exactement comme un fichier du système d'exploitation.

Le fichier d'ICE maître occupe un emplacement différent selon les systèmes d'exploitation susmentionnés. Dans certains systèmes, il existe une "variable d'environnement"¹⁸⁾ qui peut contenir le chemin conduisant à l'ICE. Dans ces cas, la variable est nommée "APPLICOM" (en majuscules). Les algorithmes suivants doivent être utilisés pour déterminer l'emplacement de l'ICE (voir le Tableau B.1).

TABLEAU B.1/T.611

Détermination de l'emplacement de l'ICE maître

Système d'exploitation	Algorithme
UNIX	Examiner d'abord le contenu de la variable "APPLICOM". S'il n'y a pas de variable, c'est que le fichier de l'ICE est placé dans le sous-répertoire /dev subdirectory et est nommé "APPLICOM_ICE"
MacOS	L'ICE est placé dans le dossier système
MS-DOS	Examiner la variable d'environnement "APPLICOM". Si cette variable n'existe pas, il y a lieu de considérer cela comme une erreur
Windows	Le fichier est appelé "ICE.INI" et est situé dans le répertoire Windows local du système d'application locale. Si ce fichier n'existe pas, examiner le fichier "WIN.INI" du répertoire Windows, section [APPLICOM_ICE]

Dans un environnement de calculateurs en réseau de zone locale (RZL), il y a lieu que l'administrateur du réseau installe le fichier d'ICE maître dans un répertoire accessible en mode partagé par toutes les applications locales possibles.

Les fichiers individuels de descripteurs d'application de communication ou de pilotes d'application de communication, dépendant d'un même utilisateur ou environnement RZL, n'ont pas besoin d'être installés dans le même répertoire que l'ICE maître; mais il convient de les placer dans des répertoires accessibles par toutes les applications locales.

Les conventions de dénomination suivantes doivent être utilisées pour désigner les fichiers de descripteurs ou de pilotes d'application de communication:

CAnnnnAC.<ext>

où

nnnn est un nombre compris dans la gamme de 1 à 9999

et

<ext> := ("ini" | "exe" | "dll" | <other>)

La chaîne "ini" doit être utilisée dans le cas d'un fichier en mode texte et la chaîne <other> peut être définie par le constructeur de l'application de communication sur la base des conventions de dénomination utilisées à l'intérieur de l'environnement de calcul visé.

L'accès au (x) descripteur (s) d'application de communication doit être défini, pour une application de communication donnée, dans le cadre de l'ICE maître, sous la forme d'un paramètre de chemin (chemin plus nom du fichier). L'emplacement par défaut du descripteur d'application de communication, si aucun chemin n'est défini, doit être le même chemin que celui de l'ICE maître.

¹⁸⁾ Les systèmes d'exploitation comme UNIX ou MS-DOS assurent une sorte d'environnement qui se compose d'un ensemble de chaînes de caractères ASCII appliquées à des variables représentées en caractères ASCII.

Annexe C

Liste des codes d'erreur APPLI/COM

(Cette annexe fait partie intégrante de la présente Recommandation)

Le Tableau C.1 indique les codes d'erreur que doivent utiliser généralement les mises en œuvre de CA conformes à la présente Recommandation.

TABLEAU C.1/T.611

Liste des codes d'erreur

Code d'erreur ^{a)}	Texte d'erreur	Observations
0000	Satisfaisant	Sans erreur – fonctionnement satisfaisant
0001-4999	Usage privé	Réservés pour usage privé de la CA
5000	Erreur d'interface de CA	
5001	Service non pris en charge spécifié	
5002	Fichier "FILENAME" existe déjà	La CA ne peut créer ou écrire le fichier "FILENAME"
5003	Erreur de création "TARGET"	La CA ne peut créer ou écrire le fichier "TARGET"
5004	Erreur de lecture liste d'adresses	
5005	Erreur de lecture liste de documents	
5006	Erreur de lecture document	
5007	Erreur d'écriture document	
5008	"Code-ID" pas admis	
5009	"Connection-ID" non valable	
5010	"CA-ID" non valable	
5011-5499	Réservés	Réservés; appelle un complément d'étude
5500-5999	Usage privé	Réservés pour usage privé de la CA
6000	Erreur de syntaxe	
6001	Fonction inconnue demandée	
6002	TDD vide	
6003	En-tête APPLI/COM non valable	Longueur ou contenu non valable
6004	Erreur d'analyseur syntaxique (ligne TDD trop longue)	
6005	Erreur d'analyseur syntaxique (ligne TDD trop courte)	
6006	Occurrences multiples de mots clés	
6007	Paramètre de taille inadéquate	
6008	Mot clé ERROR manquant	
6009	Mot clé obligatoire manquant	
6010	Mot clé non défini	
6011	Mots clés contradictoires	
6012	Paramètre hors gamme	
6013	Mots clés hors séquence	Ordre: FUNCTION, SUBFUNC, autres ...
6014	Séparateur de mots clés manquant	Symbole deux points (":") manquant
6015	"LA-ID" non valable	

TABLEAU C.1/T.611 (fin)

Liste des codes d'erreur

Code d'erreur ^{a)}	Texte d'erreur	Observations
6016	"REQ-ID" non valable	
6017	"COM-ID" non valable	
6018	"Service-ID" non valable	
6019	Le "Type-ID" ne s'adapte pas au "Service-ID"	
6020	Le "Conversion-ID" ne s'adapte pas au "Type-ID"	
6021	Fichier "TARGET" manquant	S'applique à COPY TDD
6022	Identificateur de code non valable	Identificateurs autorisés: "A", "B", "C", "E", "I" ou "P"
6023	Version d'interface incompatible	La version de l'en-tête APPLI/COM ne s'adapte pas
6024-6499	Réservés	Réservés; appelle un complément d'étude
6500-6999	Usage privé	Réservés pour usage privé de la CA
7000	Erreur imputable au matériel/système	
7001-7499	Réservés	Réservés; appelle un complément d'étude
7500-7999	Usage privé	Réservés pour usage privé de la CA
8000	Erreur en cours de conversion de document	
8001	Format de transfert non valable	
8002	Fin de document imprévue	
8003	Erreur d'accès au document imprévue	
8004	Erreur en cours de conversion T.61	
8005	Erreur en cours de conversion ASCII	
8006	Erreur en cours de conversion TIFF	
8007	Erreur en cours de conversion T.4	
8008	Erreur en cours de conversion T.6	
8009-8499	Réservés	Réservés; appelle un complément d'étude
8500-8999	Usage privé	Réservés pour usage privé de la CA
9000	Erreur de transmission	
9001	Communication interrompue	
9002	Destination distante occupée	
9003	Echec de l'établissement de la communication	
9006	Erreur de transmission (couche transport)	
9007	Erreur de transmission (couche session)	
9008	Communication fermée destination distante	
9009	La destination distante refuse l'appel	
9010	Destination distante incompatible	
9011	Echec de transmission – Erreur de document	
9012-9499	Réservés	Réservés; appelle un complément d'étude
9500-9999	Usage privé	Réservés pour usage privé de la CA
a) Les codes d'erreur sont indiqués en valeurs décimales.		

Annexe D

Exemples d'échanges de descriptions TDD

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

D.1 Exemple de session d'émission

Une application locale fonctionnant dans un environnement de type MS-DOS (ou Windows) veut envoyer un document (qui contient des informations graphiques), par exemple le document "c:\dtp\graphic1.tif", à un destinataire via les services de télécopie du groupe 3. La séquence des opérations que l'application locale (LA) doit effectuer est la suivante:

- rechercher une application de communication qui soit capable d'assurer le service de télécopie du groupe 3 en inspectant l'ICE et, si elle en trouve une, ouvrir une session avec cette application de communication par un appel de fonction d'ouverture de session (si cela n'est pas déjà fait);
- préparer le document en tant que fichier APPLI/COM TIFF (si ce n'est pas déjà fait);
- créer une description TDD de type SEND;
- transférer les descriptions TDD à l'application de communication;
- interroger régulièrement l'application de communication (ou attendre d'être informée par la fonction alarme) jusqu'à ce que les descriptions TDD de réponse soient disponibles;
- consulter les descriptions TDD de réponse pour connaître l'état de l'émission;
- fermer la session avec l'application de communication (ou effectuer d'autres fonctions avec la même application de communication).

Admettons que l'application locale (LA) ait déjà ouvert une session avec l'application de communication et que le fichier "c:\dtp\graphic1.tif" soit déjà préparé dans le format TIFF. L'application locale (LA) doit alors préparer dans sa mémoire les descriptions TDD de type SEND. L'application locale utilise le codage par défaut (T.50) pour la préparation de ces descriptions TDD, qui pourront se présenter comme indiqué à la Figure D.1.

!*APPLI/COM*1994*UIT-T*		
; Envoi d'un document graphique par télécopie du groupe 3		
; Les champs dans lesquels une valeur peut être indiquée par retour sont pré-réglés par des caractères de ; soulignement (5F _{HEX}).		
FUNCTION	: SendAck	; Envoi avec réponse
LA-ID	: myLA	; Nom de l'application locale (LA)
REQ-ID	: g_0815	; Identificateur de demande, produit par ; l'application locale
SERVICE	: FX3	; Service de télécopie groupe 3
ADDRESS	: 08154711	; Destinataire
FILENAME	: c:\dtp\graphic1.tif	; Chemin d'accès complet au document
CONVERT	: TIFF	; Format de transfert
COMID	: _____	; Identificateur unique de l'application de ; communication (réponse)
STATUS	: _____	; Etat de l'émission (réponse)
ERROR	: _____	; Est-ce qu'une erreur s'est produite? (réponse)

FIGURE D.1/T.611

Une fois que les descriptions TDD ont été préparées, l'application locale (LA) lance un appel interne à la fonction PutTDD du mécanisme d'échange de base qui transfère les descriptions TDD à l'application de communication.

Puis l'application locale (LA) interroge régulièrement l'application de communication en utilisant la fonction PollTDD du mécanisme d'échange de base jusqu'à ce que les descriptions TDD de réponse soient disponibles. En utilisant la fonction GetTDD du mécanisme d'échange de base, l'application locale (LA) consulte les descriptions TDD émanant de l'application de communication pour les placer dans sa propre mémoire puis inspecte les résultats. Les descriptions TDD de réponse peuvent se présenter comme indiqué à la Figure D.2.

```

!*APPLI/COM*1994*UIT-T*

; Envoi d'un document graphique par télécopie du groupe 3

; Les champs dans lesquels une valeur peut être indiquée par retour sont pré-réglés par des caractères de
; soulignement (5FHEX).

FUNCTION      :   SendAck                               ; Envoi avec réponse
LA-ID         :   myLA                                  ; Nom de l'application locale (LA)
REQ-ID        :   g_0815                                ; Identificateur de demande, produit par
; l'application locale
SERVICE      :   FX3                                  ; Service de télécopie groupe 3
ADDRESS       :   0498154711                            ; Destinataire
FILENAME      :   c:\dtp\graphic1.tif                   ; Chemin d'accès complet au document
CONVERT       :   TIFF                                  ; Format de transfert
COMID         :   123456                                ; Identificateur unique de l'application de
; communication (réponse)
STATUS        :   +                                     ; Etat de l'émission (réponse)
ERROR         :   0000/Success                           ; Est-ce qu'une erreur s'est produite? (réponse)

```

FIGURE D.2/T.611

Comme on peut le voir d'après les champs STATUS et ERROR, l'émission a été efficace.

D.2 Exemple de session de réception

Une application locale fonctionnant dans un environnement d'exploitation quelconque veut savoir s'il y a des documents à recevoir dans une application de communication par télécopie. La séquence des opérations que l'application locale (LA) doit effectuer est la suivante:

- rechercher une application de communication qui soit capable d'assurer le service de télécopie en inspectant l'ICE et, si elle en trouve une, ouvrir une session avec cette application de communication par un appel de fonction d'ouverture de session;
- préparer des descriptions TDD de type RECEIVE et les transférer à l'application de communication en utilisant la fonction PutTDD;
- interroger régulièrement l'application de communication (en utilisant la fonction PollTDD) (ou attendre d'être informée par la fonction alarme de rappel) jusqu'à ce que les descriptions TDD de réponse soient disponibles;
- consulter les descriptions TDD de réponse (en utilisant la fonction GetTDD) pour connaître l'état de la réception;
- fermer la session avec l'application de communication ou effectuer d'autres fonctions.

L'application locale utilise le codage par défaut (T.50) pour la préparation des descriptions TDD de type receive. Ces descriptions TDD peuvent se présenter comme indiqué à la Figure D.3.

```

!*APPLI/COM*1994*UIT-T*

; Envoi d'un document en mode T.4 par télécopie du groupe 3

; Les champs dans lesquels une valeur peut être indiquée par retour sont pré-réglés par des caractères de
; soulignemen (5FHEX).

FUNCTION      :   Receive                               ; Envoi avec réponse
LA-ID         :   myLA                                 ; Nom de l'application locale (LA)
REQ-ID        :   g_0816                               ; Identificateur de demande, produit par
; l'application locale
SERVICE      :   FX3                                  ; Service de télécopie groupe 3
FILENAME      :   c:\file.ext                          ; Chemin d'accès complet au document
Cvfax3        :   TIFF                                 ; Format de transfert désiré
TypeID        :   _____                          ; Etat du fichier reçu
ADDRESS       :   _____                          ; Adresse de l'expéditeur,
; remplie par CA
CONVERT       :   _____                          ; Format de transfert
COMID         :   _____                          ; Identificateur unique de CA (réponse)
STATUS        :   _____                          ; Etat de l'émission (réponse)
ERROR         :   _____                          ; Erreur produite? (réponse)

```

FIGURE D.3/T.611

Une fois que les descriptions TDD ont été préparées, l'application locale (LA) lance un appel interne à la fonction PutTDD du mécanisme d'échange de base qui transfère les descriptions TDD à l'application de communication.

Puis l'application locale (LA) interroge régulièrement l'application de communication en utilisant la fonction PollTDD du mécanisme d'échange de base jusqu'à ce que les descriptions TDD de réponse soient disponibles. En utilisant la fonction GetTDD du mécanisme d'échange de base, l'application locale (LA) consulte les descriptions TDD émanant de l'application de communication pour les placer dans sa propre mémoire puis inspecte les résultats. Les descriptions TDD de réponse peuvent se présenter comme indiqué à la Figure D.4.

!*APPLI/COM*1994*UIT-T*		
; Envoi d'un document en mode T.4 par télécopie du groupe 3		
; Les champs dans lesquels une valeur peut être indiquée par retour sont pré-réglés par des caractères de soulignement (5F _{HEX}).		
FUNCTION	: Receive	; Envoi avec réponse
LA-ID	: myLA	; Nom de l'application locale (LA)
REQ-ID	: g_0816	; Identificateur de demande, produit par l'application locale
SERVICE	: FX3	; Service de télécopie groupe 3
FILENAME	: c:\file.ext	; Chemin d'accès complet au document
Cvfax3	: TIFF	; Format de transfert désiré
TypeID	: STD	; Etat du fichier reçu
ADDRESS	: 033145782762	; Adresse de l'expéditeur, remplie par l'application de communication
CONVERT	: T.4	; Format de transfert
COMID	: 000001	; Identificateur unique de CA (réponse)
STATUS	: +	; Etat de l'émission (réponse)
ERROR	: 0000/Success	; Erreur produite? (réponse)

FIGURE D.4/T.611

Comme on peut le voir d'après les champs STATUS et ERROR, la réception a été efficace pour l'application de communication.

D.3 Exemple de session de suivi (Trace)

Une application locale fonctionnant dans un environnement d'exploitation peut obtenir des renseignements sur l'état transitoire ou définitif d'un enregistrement de communication (enregistrement CA). La séquence des opérations que l'application locale (LA) doit effectuer est la suivante:

- rechercher une application de communication qui soit capable d'assurer n'importe quel service de télécommunication en inspectant l'ICE et, si elle en trouve une, ouvrir une session avec cette application de communication par un appel de fonction d'ouverture de session;
- préparer des descriptions TDD de type Trace et les transférer à l'application de communication en utilisant la fonction PutTDD;
- interroger régulièrement l'application de communication (en utilisant la fonction PollTDD) (ou attendre d'être informée par la fonction alarme de rappel) jusqu'à ce que les descriptions TDD de réponse soient disponibles;
- consulter les descriptions TDD de réponse (en utilisant la fonction GetTDD) pour connaître l'état de la réception;
- fermer la session avec l'application de communication ou effectuer d'autres fonctions.

L'application locale utilise le codage par défaut (T.50) pour la préparation des descriptions TDD de type receive. Ces descriptions TDD peuvent se présenter comme indiqué à la Figure D.5.

```

!*APPLI/COM*1994*UIT-T*

; Consulter une liste de tous les enregistrements CA dans l'état "en cours d'émission"

; Les champs dans lesquels une valeur peut être indiquée par retour sont pré-réglés par des caractères de
; soulignement (5FHEX).

FUNCTION      :   Copy                               ; Emission avec réponse
LA-ID        :   myLA                               ; Nom de l'application locale
REQ-ID       :   g_0816                             ; Identificateur de demande, créé par l'application ;
                                                    locale
State        :   sending                             ; Enregistrements CA en cours de traitement
Target       :   c:\file.ext                         ; Chemin d'accès complet au document
ERROR        :   _____                         ; Erreur produite? (réponse)

```

FIGURE D.5/T.611

Une fois que les descriptions TDD ont été préparées, l'application locale (LA) lance un appel interne à la fonction PutTDD du mécanisme d'échange de base qui transfère les descriptions TDD à l'application de communication.

Puis l'application locale (LA) interroge régulièrement l'application de communication en utilisant la fonction PollTDD du mécanisme d'échange de base jusqu'à ce que les descriptions TDD de réponse soient disponibles. En utilisant la fonction GetTDD du mécanisme d'échange de base, l'application locale (LA) consulte tous les enregistrements CA émanant de l'application de communication et se trouvant dans un état donné. Ces descriptions TDD de réponse peuvent se présenter comme indiqué à la Figure D.6.

```

!*APPLI/COM*1994*UIT-T*

; Consulter une liste de tous les enregistrements CA dans l'état "en cours d'émission"

; Les champs dans lesquels une valeur peut être indiquée par retour sont pré-réglés par des caractères de
; soulignement (5FHEX).

FUNCTION      :   Copy                               ; Emission avec réponse
LA-ID        :   myLA                               ; Nom de l'application locale
REQ-ID       :   g_0816                             ; Identificateur de demande, créé par l'application ;
                                                    locale
State        :   sending                             ; Enregistrements CA en cours de traitement
Target       :   c:\file.ext                         ; Chemin d'accès complet au document
ERROR        :   0000/Success                       ; Erreur produite? (réponse)

```

FIGURE D.6/T.611

Comme on peut le voir d'après le champ ERROR, la réception a été efficace pour l'application de communication.

Annexe E

Exemple d'environnement de configuration d'interface (ICE)

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

On trouvera à la Figure E.1 un exemple d'environnement de configuration d'interface (ICE) maître, situé sous forme de fichier dans un ordinateur sous MS-DOS. Cet exemple représente un ICE maître donnant accès à plusieurs applications de communication assurant plusieurs services et méthodes d'accès.

```
I*APPLI/COM*1994*UIT-T*ICE MAÎTRE

; Noter qu'une ou plusieurs applications de communication peuvent être citées dans l'ICE maître et que chacune
; de ces applications doit inclure les mots clés APPLICOM; SERVICE; EM et ACCESS.

; Chaque nouvelle entrée de CA est signalée par un caractère "#" (signe numéro) comme indiqué ci-dessous:

#                               ; Début d'une nouvelle entrée de CA
APPLICOM:      Produit1 (c) par la société DonaldDuck ; Produit d'application de communication
                ; et fabricant
SERVICE:      FX3                               ; Assure le service de télécopie du groupe 3
SERVICE:      EMAIL                             ; Assure le service EMAIL
EM:            fichier                            ; Méthode d'échange de fichiers
ACCESS:        CA1AC.INI                          ; Descripteur de CA utilisant un fichier

#                               ; Début d'un nouvel en-tête de CA
APPLICOM:      Produit2 (c) par la société MickeyMouse ; Produit d'application de communication et
                ; fabricant
SERVICE:      FX3                               ; Assure le service de télécopie du groupe 3
SERVICE:      TLX                               ; Assure le service télex
SERVICE:      FT                               ; Assure le service de transfert de fichiers
EM:            primitive                          ; Méthode d'échange de primitives
CA-ID:         007                               ; ID de CA
ACCESS:        CA2AC.INI                          ; Descripteur de CA utilisant un fichier
ACCESS:        CA2AC.EXE                          ; Descripteur de CA exécutable
ACCESS:        CA2AC.DLL                          ; DLL de descripteur de CA
```

FIGURE E.1/T.611

Exemple d'ICE maître

On trouvera à la Figure E.2 des exemples de descripteurs de CA, situés dans un fichier dans un ordinateur sous MS-DOS.

```

I*APPLI/COM*1994*UIT-T*ICE
; Noter que n'importe quel type d'information peut être mis en mémoire dans ce cadre. Une nouvelle information
; de configuration est toujours introduite par un caractère "#" (signe numéro) comme indiqué ci-dessous.

# ; Début d'une nouvelle configuration
APPLICOM:      Produit1 (c) par la société XYZ ; Produit d'application de communication et fabricant
FC:            A ; Classe fonctionnelle APPLI/COM
EM:            FILE ; Méthode d'échange de descriptions TDD
SYNC:          No ; Application non pilotée «sync»
CODING:        I ; Codage des descriptions TDD
F_JOB_Q:       c:\applicom\job ; File d'attente de tâches
F_ACK_Q:       c:\applicom\ack ; File d'attente d'accusés de réception (réponse)
ERROR_Q:       c:\applicom\err ; File d'attente d'erreurs (réponse)
TLX:           STD ; Service télex (sans dispositif de dialogue) assuré
TX:            STD ; Service télex via télétex assuré
TTX:           STD ; Options service et type télétex
TTX:           OPD ; Options service et type télétex
TTX:           CTL ; Options service et type télétex
TTX:           DTM ; Options service et type télétex
TTX:           EDI ; Options service et type télétex
FX3:           STD ; Service de télécopie du groupe 3
ADDKEYS:       LASTTIME ; Mots clés additifs
ADDKEYS:       SUBADDR ; Mots clés additifs

# ; Début d'une nouvelle configuration
APPLICOM:      Produit2 (c) par la société ABC ; Produit d'application de communication et
; fabricant
FC:            B ; Classe fonctionnelle APPLI/COM
EM:            primitive ; Méthode d'échange de descriptions TDD
ALARM:         yes ; Rappel assuré
CODING:        I ; Codage des descriptions TDD
DRIVER:        applicom ; Module de pilotage fourni
FX3:           STD ; Options service et type télécopie G3
FX3:           DTM ; Options service et type télécopie G3
FX3:           BFT ; Options service et type télécopie G3
SUBMIT:        CONVERT ; Assure les soumissions de conversion
CONVCHK:       TIFF2 ; Des conversions en/de format TIFF2
; peuvent être demandées
CONVCHK:       PCX ; Des conversions en/de format PCX
; peuvent être demandées
ADDKEYS:       LASTTIME ; Mots clés additifs
ADDKEYS:       SPEED ; Mots clés additifs
ADDKEYS:       COMMENT ; Mots clés additifs
...

```

FIGURE E.2/T.611
Exemples de descripteurs de CA

Annexe F

Méthode d'échange dans la version 1992

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

Cette annexe énumère les fonctions mises en œuvre par la méthode d'échange de la version 1992, qui sont remplacées par les nouvelles fonctions de la méthode dite d'échange de base. Le Tableau F.1 décrit ces fonctions.

TABLEAU F.1/T.611

Comparaison des méthodes d'échange

Fonctions de la méthode d'échange (selon la version 1992)	Fonctions de la méthode d'échange de base (selon la présente Recommandation)
Login ()	ELogin ()
PutTDD ()	EPutTDD ()
-	EPutData ()
PollTDD ()	EPollTDD ()
GetTDD ()	EGetTDD ()
-	EGetData ()
SetAlarm ()	ESetAlarm ()
-	EAbortData ()
Logout ()	ELogout ()

F.1 Fonctions de la méthode d'échange selon la version 1992

F.1.1 Fonction Login (ouverture de session)

La fonction Login doit être assurée par l'application de communication. Elle doit être appelée par l'application locale avant tout échange LA-CA de demandes et de réponses concernant des descriptions TDD.

F.1.1.1 Objet

La fonction Login renvoie à l'application locale un identificateur de connexion qui sera utilisé pendant toute la durée de l'interaction LA-CA jusqu'à la fermeture de la session par l'application locale.

F.1.1.2 Comportement

L'application de communication vérifie les paramètres de l'appel d'ouverture de session. S'ils sont conformes, elle émet un identificateur de connexion que l'application locale devra utiliser ultérieurement dans les appels liés aux fonctions PutTDD, PollTDD, GetTDD et Logout. L'application locale devra attendre que l'état soit indiqué par retour pour continuer la procédure. Si l'identificateur de connexion renvoyé a la valeur Null (zéro), cela signifie que l'application de communication n'a pu connecter l'application locale pour des raisons d'identification.

F.1.1.3 Paramètres

Les paramètres suivants sont nécessaires (voir le Tableau F.2).

TABLEAU F.2/T.611

Paramètres de la fonction Login

Nom de paramètre	Structure	Commentaire	Sens
Login-Name (Nom d'ouverture de session)	String (chaîne)	Indique le nom de l'utilisateur LA utilisé pour connecter une LA à une CA (diffère de la LA-ID)	Paramètre d'entrée
Password (Mot de passe)	String (chaîne)	La LA donne son mot de passe afin que la CA puisse l'identifier	Paramètre d'entrée
Connection-ID (Identificateur de connexion)	Integer (entier)	Retourné par la CA si la fonction Login a réussi. Sinon (par exemple si l'identification a échoué) la CA met la valeur à NULL et le code d'erreur est donné par le paramètre Status (voir ci-dessous)	Paramètre de sortie
Status (Descripteur d'état)	Integer (entier)	Retourne le code d'erreur (0000 signifie succès)	Paramètre de sortie

F.1.2 Fonction PutTDD (demande de description TDD)

La fonction PutTDD doit être assurée par l'application de communication. Elle peut être invoquée par une application locale.

F.1.2.1 Objet

Le but de la fonction PutTDD est de transmettre une demande de description TDD d'une application locale à une application de communication.

F.1.2.2 Comportement

L'application de communication copie la demande de description TDD transmise par la fonction PutTDD dans ses structures internes. Le résultat se compose d'un descripteur d'état notifié immédiatement à l'application locale ayant adressé la demande et d'une référence d'identification de la demande: REQ-ID.

Les descriptions TDD sont alors analysées par l'application de communication, qui continuera à les traiter selon leur nature.

F.1.2.3 Paramètres

Les paramètres suivants sont nécessaires (voir le Tableau F.3).

TABLEAU F.3/T.611

Paramètres de la fonction PutTDD

Nom du paramètre	Structure	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	L'identificateur de connexion retourné par la fonction Login	Paramètre d'entrée
TDD location (Emplacement des TDD)	Memory address (adresse mémoire)	Spécifie où se trouve la TDD de la LA afin que la CA puisse la copier dans sa propre structure interne. Lorsque la fonction est terminée, la TDD de la LA peut être supprimée ou utilisée à d'autres fins	Paramètre d'entrée
TDD size (Taille des TDD)	Integer (entier)	Indique la taille de la TDD afin que la CA puisse attribuer suffisamment de ressources internes pour la traiter	Paramètre d'entrée
Status (Descripteur d'état)	Integer (entier)	Accuse réception de la fonction PutTDD. Retourne le code d'erreur (0000 signifie succès)	Paramètre de sortie

F.1.3 Fonction PollTDD (interrogation concernant les descriptions TDD)

La fonction PollTDD demande à l'application de communication combien de réponses contenant des descriptions TDD attendent d'être traitées par l'application locale ayant adressé la demande. La fonction PollTDD retourne le nombre de réponses TDD en attente ainsi que le type et la taille de la première réponse contenant des descriptions TDD qui sera renvoyée par le prochain appel à la fonction GetTDD.

F.1.3.1 Objet

La fonction PollTDD a pour but de préparer l'application locale pour le traitement d'une éventuelle réponse contenant des descriptions TDD émanant d'une application de communication. Elle donne également une indication du nombre de réponses contenant des descriptions TDD qui attendent d'être traitées par l'application locale.

F.1.3.2 Comportement

Lorsque l'application de communication dispose de nombreuses réponses contenant des descriptions TDD, elle choisit celle qu'elle renverra en premier. Cette réponse contenant des descriptions TDD constitue les descriptions TDD qui seront transmises à l'application locale lors du prochain appel de la fonction GetTDD émis par cette même application locale.

Lorsque aucune réponse contenant des descriptions TDD n'est disponible pour l'application locale ayant adressé la demande, le descripteur d'état renvoyé est fixé sur la valeur zéro. Si les descriptions TDD qui ont été renvoyées sont qualifiées d'erronées ou d'inconnues, par exemple si l'en-tête <TDD Header> manque, la fonction renvoie la demande de description TDD originale et met à zéro le descripteur d'état renvoyé.

Lorsqu'une réponse contenant des descriptions TDD est disponible, l'application locale doit réserver un ensemble de descriptions TDD vide qui gardera copie de la réponse TDD restée sous le contrôle de l'application de communication. Le Tableau F.4 décrit les divers types définis de description TDD.

TABLEAU F.4/T.611

Affectation des types de description TDD

Nombre	Type de description TDD	Nombre	Type de description TDD
0	Des descriptions TDD erronées ou inconnues ont été soumises – ou bien aucune réponse TDD n'a été reçue	4	Réponse de type SUBMIT
1	Réponse de type SEND	5	Réponse de type EXTEND
2	Réponse de type RECEIVE	6	Réponse de type NATIONAL
3	Réponse de type TRACE	7	Réponse de type PRIVATE

F.1.3.3 Paramètres

Les paramètres suivants sont nécessaires (voir le Tableau F.5).

F.1.4 Fonction GetTDD (consultation de description TDD)

La fonction GetTDD doit être assurée par l'application de communication. Elle peut être invoquée par une application locale.

F.1.4.1 Objet

Le but de la fonction GetTDD est de consulter une réponse contenant des descriptions TDD émanant d'une application de communication. L'application de communication copie la réponse contenant des descriptions TDD dans une structure interne de l'application locale.

TABLEAU F.5/T.611

Paramètres de la fonction PollTDD

Nom de paramètre	Structure	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	L'identificateur de connexion retourné par la fonction Login	Paramètre d'entrée
TDD size (Taille des TDD)	Integer (entier)	Indique la taille de la prochaine réponse TDD afin que la LA puisse préparer suffisamment de ressources pour la traiter	Paramètre de sortie
TDD type (Type de TDD)	Integer (entier)	Indique quel type de TDD la LA doit recevoir ensuite. Voir les valeurs ci-dessus	Paramètre de sortie
TDD count (Décompte de descriptions TDD)	Integer (entier)	Indique le nombre de réponses TDD qui sont en attente pour être récupérées par la LA. NULL signifie qu'aucune TDD n'est en attente	Paramètre de sortie
Status (Descripteur d'état)	Integer (entier)	Accuse réception de la fonction PollTDD. Retourne le code d'erreur (0000 signifie succès)	Paramètre de sortie

F.1.4.2 Comportement

L'application locale indique l'emplacement d'un ensemble de descriptions TDD vide où l'application de communication doit copier une réponse contenant des descriptions TDD disponible pour l'application locale.

L'application de communication doit renvoyer à l'application locale la réponse contenant des descriptions TDD qui a été désignée par la précédente fonction PollTDD émise par la même application locale. Celle-ci doit avoir ménagé dans ses structures internes une zone de réponses TDD du destinataire. L'invocation d'une fonction GetTDD par une application locale doit toujours être précédée de l'appel d'une fonction PollTDD.

Si l'application locale effectue deux ou plusieurs invocations consécutives de la fonction GetTDD (sans appel intermédiaire à la fonction PollTDD), les mêmes descriptions TDD seront toujours renvoyées (il s'agira des descriptions TDD indiquées par le dernier retour de la fonction PollTDD).

F.1.4.3 Paramètres

Les paramètres suivants sont nécessaires (voir le Tableau F.6).

TABLEAU F.6/T.611

Paramètres de la fonction GetTDD

Nom de paramètre	Structure	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	L'identificateur de connexion retourné par la fonction Login	Paramètre d'entrée
TDD location (Emplacement des TDD)	Memory address (adresse mémoire)	Spécifie où la CA peut copier la réponse TDD dans la structure interne de la LA. Lorsque la fonction est terminée, la réponse TDD de la CA peut être supprimée ou utilisée à d'autres fins	Paramètre d'entrée/ sortie
TDD size (Taille des TDD)	Integer (entier)	Indique la taille de la TDD vide transmise à la CA	Paramètre d'entrée
Status (Descripteur d'état)	Integer (entier)	Accuse réception de la fonction GetTDD. Retourne le code d'erreur (0000 signifie succès)	Paramètre de sortie

F.1.5 Fonction SetAlarm (alarme de consigne)

La fonction SetAlarm peut être assurée à titre facultatif par l'application de communication. Elle peut être invoquée à titre facultatif par une application locale. Si la fonction SetAlarm est utilisée par une application locale, celle-ci doit assurer la fonction CallbackRoutine. La mise en œuvre de la fonction SetAlarm est déclarée dans l'ICE.

F.1.5.1 Objet

Le but de la fonction SetAlarm est de déclarer à l'application de communication le point d'entrée de la fonction CallbackRoutine. La fonction SetAlarm indique à l'application de communication qu'elle peut éveiller l'application locale en appelant la fonction CallbackRoutine. Cette fonction peut être appelée une seule fois pendant une session de dialogue LA/CA.

F.1.5.2 Comportement

L'application de communication doit enregistrer l'emplacement de la fonction CallbackRoutine assigné par l'application locale. L'application de communication peut enregistrer autant d'emplacements de fonctions CallbackRoutine qu'il y a d'applications locales en session ouverte avec elle. L'application de communication peut alors éveiller une application locale particulière en appelant sa fonction CallbackRoutine.

F.1.5.3 Paramètres

Les paramètres suivants sont nécessaires (voir le Tableau F.7).

TABLEAU F.7/T.611

Paramètres de la fonction SetAlarm

Nom de paramètre	Structure	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	L'identificateur de connexion retourné par la fonction Login	Paramètre d'entrée
CallbackRoutine location	Memory address (adresse mémoire)	Spécifie le point d'entrée d'une fonction CallbackRoutine d'une LA spécifique	Paramètre d'entrée
Status (Descripteur d'état)	Integer (entier)	Accuse réception de la fonction SetAlarm. Retourne le code d'erreur (0000 signifie succès)	Paramètre de sortie

F.1.6 Fonction CallbackRoutine

La fonction CallbackRoutine peut être assurée à titre facultatif par l'application locale. Elle peut être appelée à titre facultatif par une application de communication. Pour ce faire, l'application locale doit l'avoir déclarée à l'application de communication au moyen de la fonction SetAlarm.

F.1.6.1 Objet

La fonction CallbackRoutine définit un mécanisme qui permet à une application de communication d'avertir l'application locale que quelques réponses contenant des descriptions TDD sont disponibles. L'utilisation de ce mécanisme facultatif peut améliorer la commande de flux entre LA et CA dans des systèmes fortement chargés.

L'invocation de la fonction CallbackRoutine par une application de communication ne garantit pas que l'application locale interrogera l'application de communication dans un certain délai. Elle garantit seulement que l'application de communication recevra une alarme d'une application de communication particulière.

L'application de communication peut répéter l'invocation de la fonction CallbackRoutine d'une application locale donnée si celle-ci n'interroge pas l'application de communication assez rapidement.

Il n'y a qu'une seule adresse de rappel autorisée par l'application locale. Si toutefois une application locale souhaite être appelée en fonction d'un événement spécifique, cette application locale doit mettre en œuvre elle-même cet élément de service, dans le cadre de la fonction de rappel invoquée.

F.1.6.2 Comportement

La fonction CallbackRoutine a pour but de permettre à une application de communication en session ouverte d'avertir l'application locale que l'application de communication doit être interrogée au moyen de la fonction PollTDD. Ainsi, l'application locale devra interroger l'application de communication dès que possible afin qu'une situation de débordement, entraînant un état d'erreur, ne se produise dans l'application de communication.

F.1.6.3 Paramètre

Le paramètre suivant est nécessaire (voir le Tableau F.8).

TABLEAU F.8/T.611

Paramètre de la fonction CallbackRoutine

Nom de paramètre	Structure	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	L'identificateur de connexion retourné par la fonction Login	Paramètre d'entrée

F.1.7 Fonction Logout (fermeture de session)

La fonction Logout doit être assurée par l'application de communication. Elle doit être appelée par l'application locale à la fin de tout échange LA-CA de demandes et de réponses contenant des descriptions TDD.

F.1.7.1 Objet

La fonction Logout renvoie à l'application locale un descripteur d'état qui indique si l'interaction LA-CA a pris fin dans les conditions normales.

F.1.7.2 Comportement

Avant de mettre fin au dialogue LA-CA, l'application de communication peut (mais sans y être tenue) traiter toutes les demandes de descriptions TDD en suspens qui ont été émises par cette application locale.

F.1.7.3 Paramètres

Les paramètres suivants sont nécessaires (voir le Tableau F.9).

TABLEAU F.9/T.611

Paramètres de la fonction Logout

Nom de paramètre	Structure	Commentaire	Sens
Connection-ID (Identificateur de connexion)	Integer (entier)	L'identificateur de connexion retourné par la fonction Login	Paramètre d'entrée
Status (Descripteur d'état)	Integer (entier)	Retourné par la CA. Indique si le Logout s'est déroulé correctement. Retourne le code d'erreur (0000 signifie succès)	Paramètre de sortie

Annexe G

Informations spécifiques des services

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

G.1 Ligne d'identification d'appel (CIL) des services de télécopie G4 et de télétext

La ligne d'identification d'appel (CIL) est définie par la Recommandation UIT-T F.200 et se présente comme indiqué à la Figure G.1.

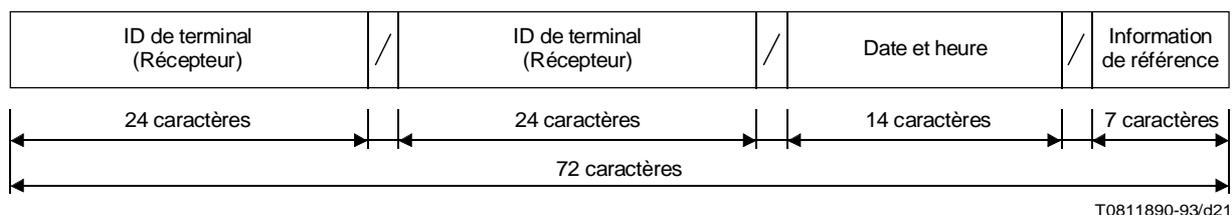


FIGURE G.1/T.611

Présentation de la ligne d'identification d'appel; "/" est le code du caractère 2F_{HEX}

G.2 Identificateur de terminal

L'identificateur de terminal est défini par la Recommandation UIT-T F.200 pour le service télétext et par la Recommandation UIT-T F.184 pour le service de télécopie du groupe 4. Cet identificateur se présente comme indiqué à la Figure G.2. L'abréviation DNIC correspond aux termes Data Network Identification Code (code d'identification de réseau pour données).

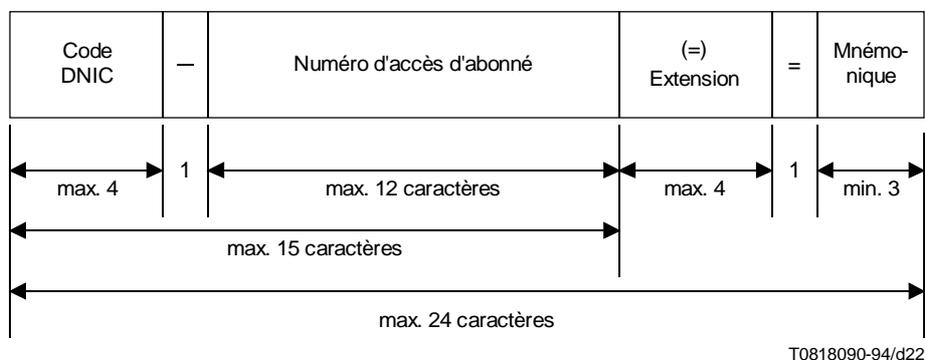


FIGURE G.2/T.611

Identificateur de terminal de télécopie G4/de télétext

Annexe H

Relevé des formats de transfert et d'émission

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

H.1 Formats de transfert en fonction des formats d'émission

Voir le Tableau H.1

TABLEAU H.1/T.611

Résumé des formats de transfert

Paramètre d'identification du service	Paramètre d'identification du type de document	Paramètre d'identification de conversion autorisé pour	
		des documents sortants	des documents entrants
FX3	STD	ASCII, ASCII _{xxx} ^{a)} , T.50, TIFF	TIFF, TIFF _x ^{b)}
	BTM, DTM, BFT, EDI	VOID	
FX4	STD	ASCII, ASCII _{xxx} , T.50, TIFF	TIFF, TIFF _x
	DTM, BFT, EDI	VOID	
TLX	STD	ASCII, ASCII _{xxx} , T.50, T.61	
TX	STD	ASCII, ASCII _{xxx} , T.50, T.61	
TTX	STD	ASCII, ASCII _{xxx} , T.50, T.61	
	OPD, MD, CTL	ASCII, ASCII _{xxx} , T.50, T.61	
	DTM, BFT, EDI	VOID	
EMAIL	STD	T.50	
	TELETEX	ASCII, ASCII _{xxx} , T.50, T.61	
	G3FAX, G4CLASS1	ASCII, ASCII _{xxx} , T.50, TIFF	TIFF _x
	VIDEOTEX, MESSAGE, BILATERAL, NATIONAL, ODA	VOID	
FT	STD	VOID	

a) "xxx" dans l'expression "ASCII_{xxx}" correspond au numéro de la page de code (par exemple "ASCII437").

b) "x" correspond à la classe de format TIFF à lire; sa valeur peut être comprise entre 2 et 4.

H.2 Formats d'émission associés à un service

Voir le Tableau H.2

TABLEAU H.2/T.611

Relevé des formats d'émission (selon les types de document)

Identificateur de service	Identificateur de type	Signification
TLX	STD	Service télex (sans possibilité de dialogue)
TX	STD	Service télex par convertisseur TELETEX
TTX	STD ^{a)}	TELETEX de base (T.61)
	OPD ^{a)}	TELETEX de base (T.61): Operator Document (document d'opérateur)
	MD ^{a)}	TELETEX de base (T.61): Monitor Document (document de contrôle)
	CTL ^{a)}	TELETEX de base (T.61): Control Document (document de commande)
	DTM ^{b)}	Transfert télématique de fichiers (TFT) du service TELETEX: Document Transparent Mode (mode de transfert transparent de documents)
	BFT ^{c)}	Transfert de fichiers binaires du service TELETEX
	EDI ^{b)}	Transfert télématique de fichiers (TFT) du service TELETEX: Edifact
FX3	STD	Service de télécopie de base G3 (MH)
	BTM ^{b)}	Transfert télématique de fichiers (TFT) du service de télécopie G3: Basic Transparent Mode (mode transparent de base)
	DTM ^{b)}	Transfert télématique de fichiers (TFT) du service de télécopie G3: Document Transparent Mode (mode de transfert transparent de documents)
	BFT ^{d)}	Transfert de fichiers binaires du service de télécopie G3
	EDI ^{b)}	Transfert télématique de fichiers (TFT) du service de télécopie G3: Edifact
FX4	STD	Service de télécopie G4 de base (MR)
	DTM ^{b)}	Transfert télématique de fichiers (TFT) du service de télécopie G4: Document Transparent Mode (mode de transfert transparent de documents)
	BFT ^{c)}	Transfert de fichiers binaires du service de télécopie G4
	EDI ^{b)}	Transfert télématique de fichiers (TFT) du service de télécopie G4: Edifact
EMAIL	STD	Partie corps de texte IA5
	TELETEX	Partie corps télétext
	G3FAX	Partie corps télécopie groupe 3
	G4CLASS1	Partie corps télécopie groupe 4
	VIDEOTEX	Partie corps vidéotex
	MESSAGE	La partie corps contient un message IPM (constitué d'un en-tête et de parties corps) tel que transféré par le système de courrier électronique
	BILATERAL	Contenu de partie corps défini par accord bilatéral
	NATIONAL	Contenu de partie corps défini à l'échelon national
	ODA	Partie corps ODA
FT	STD	Transfert de fichiers de base
<p>a) Type de document selon Annexe E/T.62.</p> <p>b) Transfert télématique de fichiers (TFT) selon la Recommandation T.571.</p> <p>c) Transfert de fichiers binaires selon la Recommandation T.434.</p> <p>d) Transfert de fichiers binaires selon les Recommandations T.434 et T.30 (Annexes).</p>		