



中华人民共和国国家标准

GB/T 16505.2—1996
idt ISO 8571-2:1988

信息处理系统 开放系统互连 文卷传送、访问和管理 第2部分：虚文卷存储器定义

Information processing systems—
Open Systems Interconnection—
File Transfer, Access and Management—
Part 2: Virtual filestore definition

1996-09-02 发布

1997-05-01 实施

国家技术监督局 发布

中 华 人 民 共 和 国
国 家 标 准
信息处理系统 开放系统互连
文卷传送、访问和管理
第 2 部分:虚文卷存储器定义

GB/T 16505.2—1996

*

中国标准出版社出版
北京复兴门外三里河北街 16 号
邮政编码:100045

电 话:68522112

中国标准出版社秦皇岛印刷厂印刷
新华书店北京发行所发行 各地新华书店经售
版权专有 不得翻印

*

开本 880×1230 1/16 印张 3 $\frac{3}{4}$ 字数 115 千字
1997 年 11 月第一版 1997 年 11 月第一次印刷
印数 1—800

*

书号: 155066 · 1-14151 定价 23.00 元

*

标 目 321—46

目 次

前言	Ⅲ
ISO 前言	Ⅳ
0 引言	1
1 范围	2
2 引用标准	2
3 定义	2
4 缩略语	2
第一篇 文卷存储器模型	
5 基本概念	3
6 文卷选择	4
7 文卷结构	4
8 对文卷的动作	9
9 属性	10
第二篇 文卷存储器上的动作	
10 对整个文卷的动作	11
11 文卷访问动作	12
第三篇 属性定义	
12 文卷属性	13
13 活动属性	18
14 属性组	20
15 最小属性值域	21
附录 A(标准的附录) 文卷访问结构约束集	23
附录 B(标准的附录) 文件类型	32
附录 C(提示的附录) 结构文卷的读	46
附录 D(提示的附录) 结构文卷中的插入	49
附录 E(提示的附录) ASN.1 交叉对照表	53
图	
1 文卷、属性和联系的关系	3
2 表示成树结构的访问结构	5
3 文卷内容的 ASN.1 定义	5
4 文卷结构的 ASN.1 定义	5
5 无结构文卷传送	47
6 平坦文卷传送	47
7 分级文卷传送	48
8 被传送数据的源	49

GB/T 16505.2—1996

9	宿文卷的初始状态	49
10	对 A 作姐妹节点插入动作的最终状态	49
11	对 C 作子节点插入动作的最终状态(常规)	50
12	对 C 作子节点插入动作的最终状态(变体)	51
13	插入子树作为姐妹节点的最终状态	51
14	插入子树作为子节点的最终状态	52
15	有序平坦文卷的初始状态	52
16	合并有序平坦文卷的最终状态	52

表

1	按访问上下文读的结果	8
2	动作对位置的影响	13
3	并发控制选项	19
4	活动属性	22
5	文卷属性	22
6	在无结构约束集中的基本约束	24
7	顺序平坦约束集中的基本约束	25
8	顺序平坦约束集中的标识约束	25
9	有序平坦约束集中的基本约束	26
10	在有序平坦约束集中的标识约束	27
11	具有唯一命名的有序平坦约束集中的基本约束	28
12	具有唯一命名的有序平坦约束集中的标识约束	28
13	有序分级约束集中的基本约束	29
14	有序分级约束集中的标识约束	29
15	通用分级约束集中的基本约束	30
16	通用分级约束集中的标识约束	31
17	具有唯一命名的通用分级约束集中的基本约束	32
18	具有唯一命名的通用分级约束集中的标识约束	32
19	无结构正文文件类型中的信息客体	33
20	顺序正文文卷文件类型中的信息客体	36
21	无结构二进制文件类型中的信息客体	40
22	顺序二进制文件类型中的信息客体	42
23	分级文件类型中的信息客体	46

前 言

本标准等同采用国际标准 ISO 8571-2:1988《信息处理系统 开放系统互连 文卷传送、访问和管理 第2部分:虚文卷存储器定义》,并根据 ISO 8571-2:1988/Cor. 1:1991《技术更正 1》进行了更正。

通过制定这项国家标准,有利于文卷在开放系统互连中的传送、访问和管理。

本标准与 ISO/IEC 8571-2:1988 的区别如下:

a) 正文和附录中引用其他标准时,用我国的标准编号代替相应的国际标准编号,其对应关系是:

GB 5261—94 代替 ISO 6429:1988

GB 9387—88 代替 ISO 7498:1988

GB/T 16505—1996 代替 ISO 8571:1988

GB/T 7408—94 代替 ISO 8601:1988

GB/T 15695—1995 代替 ISO 8822:1988

GB/T 16262—1996 代替 ISO 8824:1990

GB/T 16263—1996 代替 ISO 8825:1990

b) 根据编写国家标准的基本规定,本标准保留了被采用国际标准的前言,同时增加“前言”。

GB/T 16505 在《信息技术 开放系统互连 文卷传送、访问和管理》总标题下,目前包括以下 5 个部分:

——第 1 部分:概论

——第 2 部分:虚文卷存储器定义

——第 3 部分:文卷服务定义

——第 4 部分:文卷协议规范

——第 5 部分:文卷协议一致性声明形式

本标准的附录 A 和 B 是标准的附录,附录 C、D 和 E 为提示的附录。

本标准由中华人民共和国电子工业部提出。

本标准由电子工业部标准化研究所归口。

本标准起草单位:东南大学。

本标准主要起草人:龚俭、丁伟、高健。

ISO 前 言

ISO(国际标准化组织)是由各个国家标准机构(ISO 的成员体)联合组成的一个世界性组织。该组织通过其各个技术委员会进行国际标准的制定工作。凡是对于已设有技术委员会的某一专业感兴趣的每一个成员体,都有权参加该技术委员会。与 ISO 有联系的官方和非官方国际组织也可参与国际标准的制定工作。ISO 与国际电工委员会(IEC)在电子技术标准化的所有方面都进行密切合作。

各个技术委员会提出的国际标准草案,须先分发给各成员体表决通过后,再由 ISO 理事会批准为国际标准。根据 ISO 工作导则,国际标准至少需要投票成员体的 75%赞成。

国际标准 ISO 8571 是由 ISO/TC 97“信息处理系统”技术委员会制定的。

用户应随时注意引用的所有国际标准的修订,以及参考其他国际标准的最新版本,除非另有声明。

ISO 8571 在《信息处理系统 开放系统互连 文卷传送、访问和管理》总标题下,目前包括以下 5 个部分:

- 第 1 部分:概论
- 第 2 部分:虚文卷存储器定义
- 第 3 部分:文卷服务定义
- 第 4 部分:文卷协议规范
- 第 5 部分:文卷协议一致性声明形式

附录 A 和 B 都是标准的附录,附录 C、D 和 E 是提示的附录。

中华人民共和国国家标准

信息处理系统 开放系统互连 文卷传送、访问和管理 第2部分:虚文卷存储器定义

GB/T 16505.2—1996
idt ISO 8571-2:1988

Information processing systems—
Open Systems Interconnection—
File Transfer, Access and Management—
Part 2: Virtual filestore definition

0 引言

GB/T 16505 是为了便于计算机系统互连而制定的一套国家标准中的一个,开放系统互连参考模型(GB 9387)定义了它与这套国家标准中的其他标准的关系。参考模型把互连标准化的范围再细分成为一系列大小可管理的规范层。

开放系统互连的目标是把互连标准以外的技术协定减到最少,使下列计算机可以互连:

- a) 来自不同的厂商的;
- b) 在不同的管理下的;
- c) 具有不同复杂程度的;
- d) 不同的年代的。

GB/T 16505 定义文卷服务,并规定了在参考模型应用层可用的文卷协议。所定义的服务属于应用服务元素(ASE)一类。它涉及的是可作为文卷的可标识的信息体。文卷可存储在开放系统内或者在应用进程之间传递。

GB/T 16505 定义了基本的文卷服务。它提供支持文卷传送的基本业务,并建立文卷访问和文卷管理的框架。GB/T 16505 不对本地系统内的文卷传送或访问业务规定接口。

GB/T 16505 由下列 5 部分构成:

- 第1部分:概论
- 第2部分:虚文卷存储器定义
- 第3部分:文卷服务定义
- 第4部分:文卷协议规范
- 第5部分:文卷协议一致性声明形式

GB/T 16505 这一部分的定义用于 GB/T 16505 规定服务和协议的后续部分中,它们引用文卷存储器的定义以赋给它们所操作的各种描述性数据项以意义。当协议的实现者选择从协议项到实存储机制映射的时候也使用这个定义。

GB/T 16505 这一部分包括下列附录,它们是本标准的一部分:

- 附录 A:文卷访问结构约束集;
- 附录 B:文件类型。

GB/T 16505 这一部分还包括下列附录,它们不是本标准的一部分:

附录 C:结构文卷的读;
附录 D:结构文卷中的插入;
附录 E:ASN.1 交叉对照表。

1 范围

GB/T 16505 的这一部分

- a) 定义了虚文卷存储器的抽象模型用以描述文卷和文卷存储器(见第一篇);
- b) 定义了可用于操作该模型元素的动作之集合(见第二篇);
- c) 用属性定义了各个文卷和联系的性质(见第三篇);
- d) 用分级结构定义了文卷的表示形式(见第一篇第7章)。

GB/T 16505 这部分并不规定

- e) 有关从实文卷存储器到虚文卷存储器的映射要求;
- f) 实现实文卷存储器的要求。

该虚文卷存储器的定义将提供给本标准中定义文卷服务(GB/T 16505.3)和规定文卷协议(GB/T 16505.4)的其他部分引用。

2 引用标准

下列标准所包含的条文,通过在本标准中引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有标准都会被修订,使用本标准的各方应探讨使用下列标准最新版本的可能性。

GB/T 5261—94 信息处理 七位和八位编码字符集用的控制功能(idt ISO 6429:1988)

GB 9387—88 信息处理系统 开放系统互连 基本参考模型(idt ISO 7498:1988)

GB/T 16505—1996 信息处理系统 开放系统互连 文卷传送、访问和管理:

第1部分:概论(idt ISO 8571-1:1988)

第3部分:文卷服务定义(idt ISO 8571-3:1988)

第4部分:文卷协议规范(idt ISO 8571-4:1988)

第5部分:文卷协议一致性声明(idt ISO 8571-5:1988)

GB/T 7408—94 数据元素与交换格式 信息交换 日期与时间的表示(idt ISO 8601:1988)

GB/T 15695—1995 信息处理系统 开放系统互连 面向连接的表示服务定义(idt ISO 8822:1988)

GB/T 16262—1996 信息处理系统 开放系统互连 抽象语法记法一(ASN.1)的规范(idt ISO 8824:1990)

GB/T 16263—1996 信息处理系统 开放系统互连 抽象语法记法一(ASN.1)的基本编码规则规范(idt ISO 8825:1990)

ISO 8650—88 信息处理系统 开放系统互连 联系控制服务元素的协议规范

ISO 8832:1989 信息处理系统 开放系统互连 作业传送与操纵的基本类协议规范

ISO 9804:1988 信息处理系统 开放系统互连 托付、并发和恢复服务元素的服务定义

ISO 9834.2:1993 信息处理系统 指定开放系统互连注册权威机构的规程:第2部分:开放系统互连文件类型的注册

3 定义

本标准中所用的术语定义在 GB/T 16505.1 中。

4 缩略语

本标准中所用的缩略词定义在 GB/T 16505.1 中。

第一篇 文卷存储器模型

5 基本概念

虚文卷存储器由一个或多个应用实体来处理,发起者可以借助初始化一个提供文件服务的 FTAM 时期来与它们通信。在 FTAM 时期有能力作为响应者操作的每个应用实体都支持一个虚文卷存储器,于是一个应用实体标题可用于标识一个虚文卷存储器。

注 无二义的应用实体标题分配机制以及相应的提供该实体通信地址的目录服务机制均不在 GB/T 16505 范围中。

文卷存储器可以包含任意数目(大于或等于零)的文卷(见图 1)。

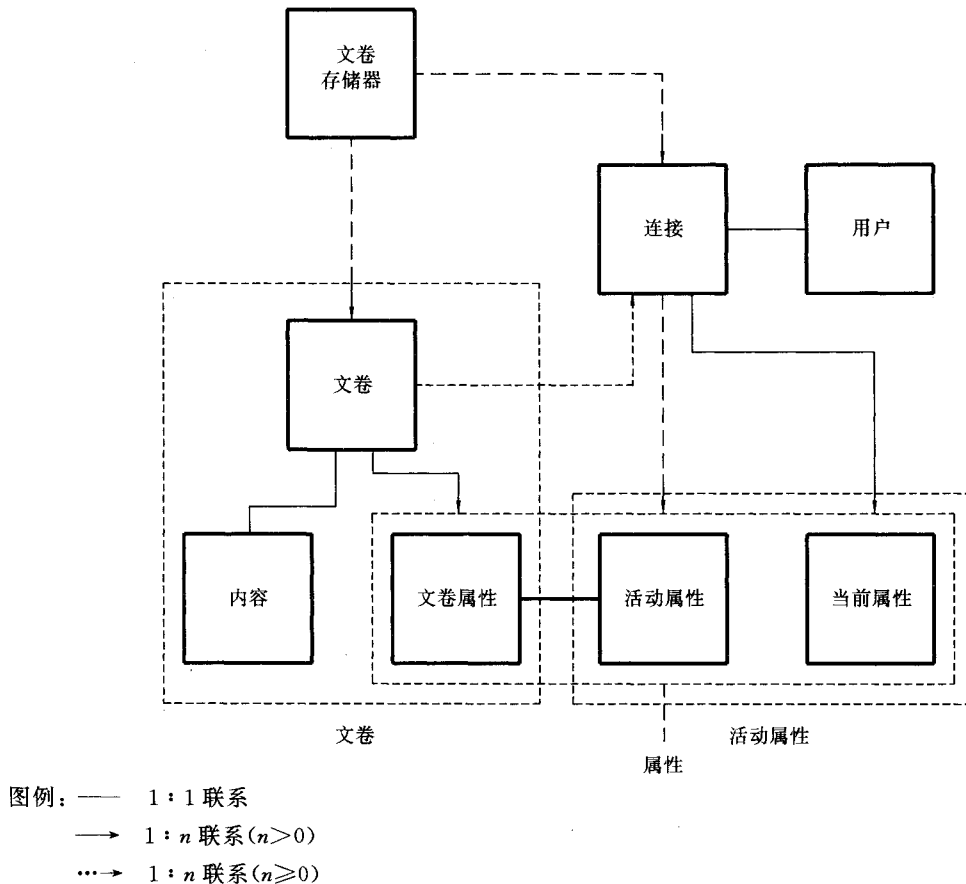


图 1 文卷、属性和联系的关系

每个文卷的性质都由一组文卷属性的值定义。这些属性是全局的；任何时候一个属性值都是对全体发起者可用的。

每个文卷不是空的就是有某种结构和一些内容,有关内容结构的各个方面由某些文卷属性标识。

每个 FTAM 时期都有一组有关的活动属性,分为两种类型。

第一种与文卷属性一一对应,并且指明由发起者所看到的那些属性的现役值。

第二种活动属性是状态信息的当前值,它涉及发生在 FTAM 时期内的交换和建立起的有关发起者的事实,通常这些信息是从协议交换的相应参数中得出。发起者只能得到他正参与的 FTAM 时期的描述。

在任何时候都可能有任何数目(大于或等于零)的发起者已经初始化了 FTAM 时期。在特定 FTAM 时期内,发起者与响应者之间的交换导致在响应者的虚文卷存储器中任何时候至多只可选择一个文卷。

6 文卷选择

在外部总是通过引用文卷名来选择文卷存储器中的文卷。对文卷的引用是在一个被应用实体标题所标识的特定文卷存储器的上下文之内的。应用实体标题与文卷存储的位置有关,它为文卷服务用户所知,但其具体不在 FTAM 的范围之内。文卷名在第 12 章中定义。

文卷的选择分两步,首先通过处理虚文卷存储器的应用实体初始化一个 FTAM 时期,然后把信息送给这个实体,以便从文卷存储器内的所有文卷中无二义性的标识该文卷。

通常,藉助于陈述给定值与文卷属性之间的一些关系也能够做出选择,然而在 GB/T 16505 总是通过文卷名来引用文卷。

7 文卷结构

7.1 文卷访问结构

本章定义分级结构的性质。在这部分,分级文卷的抽象结构用 ASN.1 定义在 ASN.1 模块 GB/T 16505-FADU 内,其中还包括平坦文卷和无结构文卷等简单情况(见 7.2)。分级结构具有下列性质:

- a) 文卷访问结构是一个有序树;
- b) 每个节点带有 0 个或 1 个数据单元;
- c) 在该结构之内,对每个节点都可访问到它的子树,该访问单元(即子树)称作文卷访问数据单元(FADU),它由该子树的结构化节点和包含在该子树内的数据单元组成,从树的根节点可访问到整个文卷;
- d) 每个节点可任选地赋予一个名字;
- e) 级的数目、弧的长度和每个节点始发弧的数目均不受限制。

注

1 适用的约束集可能会限定最大级数和允许的弧长。

2 图 2 中解释了这些术语,为清楚起见,图中每个节点已用字母从 A 到 F 唯一地标识了。

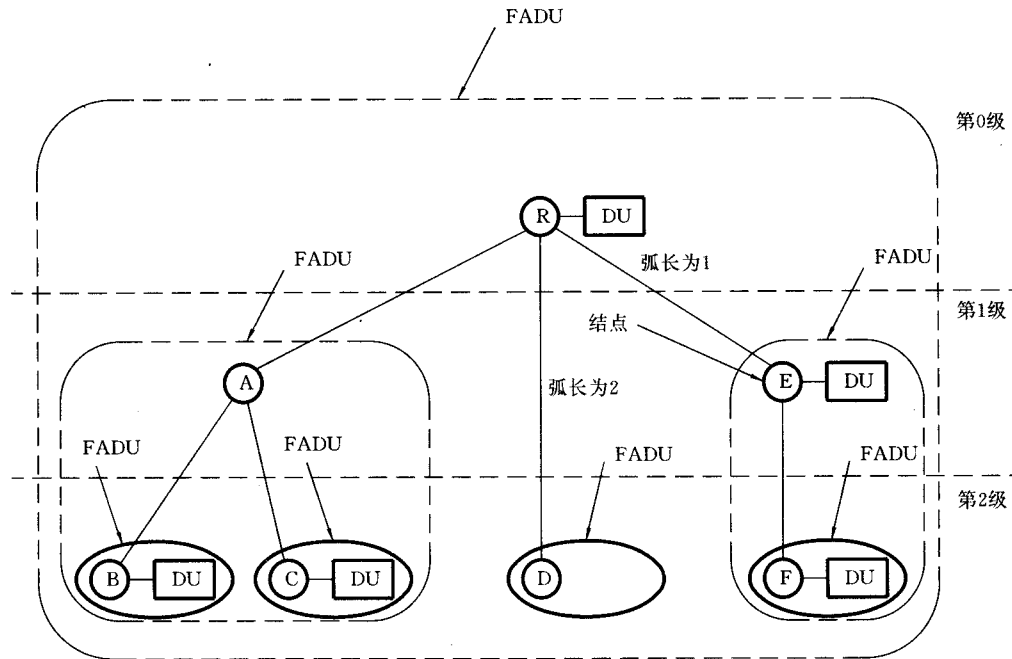
前序遍历序列是树中节点的特殊序列;它由遍历对应于整个文卷的子树来确定。遍历子树时把该子树的根节点附加到已确定的遍历序列末端上,然后按照该子树每个子节点出现的次序遍历那棵子树。

注:该算法描述了遍历序列的抽象构造,这个序列可用于建立 FADU 标识,定义定位操作和确定传输次序。

GB/T 16505 这部分并不定义如何实现该算法。

与子树一一对应的文卷访问数据单元(FADU)可按照与子树相同的方式来标识(即用它们的根节点)。类似地,数据单元总是同节点相关联的,故可以用该节点的标识来表示。通用分级结构能够表示不同范围的实际文卷结构。然而,实系统只能支持有限范围的结构,且在修改文卷的方式上也存在着限制。为了说明这些,需要引进约束集的概念。约束集定义了对允许的结构范围的限制,并且表达了基本访问动作能够修改该结构的方式。在 GB/T 16505 的这部分中定义了几个反映某些公共文卷类型的约束集(见附录 A),而另外的约束集可以随后定义和登记。

注:预期将建立登记权限机构以维护已登记的约束集。



注：图中标识了树的下列特性(子树由根节点标识)：

树的根节点：节点 R

前序遍历序列：节点 R, A, B, C, D, E, F

出现的次序：从左至右

图 2 表示成树结构的访问结构

7.2 抽象结构定义

分级文卷模型的访问结构用 ASN.1 定义在图 4 中。数据单元内容可如图 3 所示用 ASN.1 来表达，也可按某种其他抽象语法记法来表达。

1	ISO 8571-CONTENTS DEFINITIONS::=
2	
3	BEGIN
4	File-Contents-Data-Element::= ANY
5	—— 文卷内容数据元素(File-Contents-Data-Element)的值总是在表示
6	—— 上下文里传送,此表示上下文不同于用来传送 FTAM PCI 的表示上下
7	—— 文。允许的实际表示数据值从文卷内容的抽象语法中得到,该抽象
8	—— 语法规定在该文卷的内容类型文卷属性中。
9	
10	END

图 3 文卷内容的 ASN.1 定义

1	ISO 8571-FADU DEFINITIONS::=
2	
3	BEGIN
4	
5	Subtree::=SEQUENCE{
6	node Node-Descriptor-Data-Element,
7	data [0] IMPLICIT DU OPTIONAL,
8	—— 当且仅当 DU 被连接到该节点时才存在
9	children [1] IMPLICIT Children OPTIONAL}

图 4 文卷结构的 ASN.1 定义

10 ——叶节点的特征在于没有子节点
 11
 12 Children::=SEQUENCE{
 13 enter-subtree Enter-Subtree-Data-Element,
 14 SEQUENCE OF Subtree
 15 ——子树的出现次序必须遵照作为他们父节点的子节点的排序
 16 exit-subtree Exit-Subtree-Data-Element}
 17
 18 DU::=SEQUENCE OF ISO 8571-CONTENTS. File-Contents-Data-Element
 19
 20
 21 Node-Descriptor-Data-Element::=[APPLICATION 0] IMPLICIT SEQUENCE{
 22
 23 name Node-Name OPTIONAL,
 24 ——仅当子树的根节点是有名节点时才存在
 25 arc-length [1] IMPLICIT INTEGER DEFAULT 1,
 26 ——用来规定从它的父节点到子树根节点弧的长度
 27 data-exists [2] IMPLICIT BOOLEAN DEFAULT TRUE}
 28 ——data-exists=TRUE 指明 DU 被连接到该子树的根节点
 29
 30 Node-Name::=CHOICE{
 31 ftam-coded [0] IMPLICIT GraphicString,
 32 ——当采用 ftam-coded 时,节点名(Node-Name)与结构信息属于同一
 33 ——个抽象语法,故节点名在为支持 FTAM FADU 抽象语法而建立的表
 34 ——示上下文里传递。仅当内容类型文卷属性包含文件类型名时,才
 35 ——只允许节点名的这种形式,要支持这样的方法应至少支持登记
 36 ——在字符集登记项 2 中的 GO 字符集。
 37 user-coded EXTERNAL}
 38 ——允许的实际类型定义在文卷内容的抽象语法中,而该抽象语法
 39 ——由该文卷的内容类型文卷属性给出。
 40
 41 Enter-Subtree-Data-Element::=[APPLICATION 1] IMPLICIT NULL
 42
 43 Exit-Subtree-Data-Element::=[APPLICATION 2] IMPLICIT NULL
 44 ——子树入口和子树出口数据元素用来括住子树表,这些子树是该
 45 ——表先趋节点的子节点。
 46
 47 FADU::=Subtree
 48
 49 Structuring-Data-Element::=CHOICE{
 50 Node-Descriptor-Data-Element,
 51 Enter-Subtree-Data-Element,
 52 Exit-Subtree-Data-Element}
 53
 54 ——Data-Element 被定义成是通用数据类型,它的值可以是
 55 ——a)在抽象语法“FTAM FADU”里 ASN.1 类型 Structuring-Data-
 56 ——Element 的值;或者
 57 ——b)在由内容类型文件属性导出的抽象语法里 ASN.1 类型 GB/T 16505
 58 ——CONTENTS. File-Contents-Data-Element 的值。
 59
 60 END

图 4(完)

文卷按文卷访问数据单元(FADU)进行访问,它们与通用分级文卷模型中的子树等价。为访问而规定的最小数据量是一个数据单元(DU)。为了进行带检验点的传送,数据单元可划分为被称为数据元素(DE)的更小部分。不能用在 GB/T 16505 中定义的文卷存储器动作去访问一个数据单元中的某个数据元素;数据单元要作为整体访问。

7.3 抽象语法定义

为了能够在 FTAM 分级文卷模型里访问文卷结构,本标准指定 ASN.1 客体标识符值

{iso standard 8571 abstract-syntax(2) ftam-fadu(2)}

作为相应表示数据值集的抽象语法名,这些表示数据值中的每一个都是具 ASN.1 类型 ISO 8571-FADU.Structuring-Data-Element 的值。

对应的客体描述符值应为

“FTAM FADU”

ASN.1 客体标识符值

{joint-iso-ccitt asnl(1) basic-encoding(1)}

和客体描述符值

“Basic Encoding of a single ASN.1 type”

(在 GB/T 16263 中赋给信息客体)可作为这个抽象语法的传送语法名。

7.4 文卷传送结构

ASN.1 模块 ISO 8571-FADU 定义了通用分级模型的访问结构,从中可得出传送文卷内容用的语法,该语法由具有 Data-Element 类型的值构成。只有这些值才被从 FTAM 发送者借助于表示服务(GB/T 15695)传送到 FTAM 接收者。然而,这些值必须按照它们的语法次序出现。

注:语法的构造要使语法次序与前序遍历次序等价。

结构信息(即类型 Structuring-Data-Element 的值)是在 FTAM FADU 上下文(它对应于 FTAM FADU 抽象语法,见 7.3)中通信的。节点名的通信或是在相同的上下文中或是在不同的但是被嵌入的上下文中。具有类型 File-Contents-Data-Element 的值是在文卷内容表示上下文中通信的。

用来传送文卷的部分或全部内容的抽象语法是由内容类型文卷属性指明的,每个使用的抽象语法都与不同的表示上下文相关联。存在有两种可能性:

a) 内容类型文卷属性值规定一个抽象语法和一个约束集;针对规定的抽象语法需要一个表示上下文,这种表示上下文被用来传送节点名和文卷实际内容。如果使用的约束集支持文卷结构信息时,则要求有一个对应于该文卷结构抽象语法(见 7.3)的不同表示上下文。

b) 内容类型属性值规定文件类型;对于在文件类型登记项中规定的每个抽象语法都要求一个表示上下文。如果存在节点名,该文件类型还定义了要用哪个表示上下文来传送可能存在的节点名。它们可以是

- 1) 处在对应于由文件类型定义的抽象语法之一的表示上下文中(user-coded);
- 2) 与文卷结构信息处在相同的表示上下文中(ftam-coded)。

7.5 访问上下文

使用定义在 ISO 8571-FADU 模块中的抽象结构来导出供传送的相应信息序列会得到文卷完整的分级结构,也就是,在指定的 FADU 里所有的结构信息和数据都将被传送。然而,可以用不同的访问上下文从文件结构的受限角度对文卷作读访问。在所有情况下,那些被传送的数据元素处在 ISO 8571-FADU 所定义的次序中,且节点按照它们出现在前序遍历序列里的次序传送。

7.5.1 HA——分级的所有数据单元访问上下文

在 HA 访问上下文中,被寻址的 FADU 中所有四种类型的数据元素(Node-Descriptor-Data-Element, Enter-Subtree-Data-Element, Exit-Subtree-Data-Element, File-Contents-Data-Element)都要被传送。

7.5.2 HN——分级的无数据单元访问上下文

在 HN 访问上下文中,被寻址的 FADU 中类型为 Node-Descriptor-Data-Element, Enter-Subtree-Data-Element 和 Exit-Subtree-Data-Element 的所有数据元素都要被传送。

7.5.3 FA——平坦的所有数据单元访问上下文

在 FA 访问上下文中,被寻址的 FADU 中类型为 Node-Descriptor-Data-Element, 和 File-Contents-Data-Element 的数据元素都要被传送。对于那些类型为 Node-Descriptor-Data-Element 的数据元素,只有 data-exists 值为 TRUE 的才被传送。

7.5.4 FL——平坦的一级数据单元访问上下文

在 FL 访问上下文中,被寻址的 FADU 中属于规定级的那些节点的具有类型 Node-Descriptor-Data-Element 和 File-Contents-Data-Element 的数据元素都要被传送,其中对于那些类型为 Node-Descriptor-Data-Element 的数据元素,只有 data-exists 值为 TRUE 的才被传送。

7.5.5 FS——平坦单个数据单元访问上下文

在 FS 访问上下文中,属于被寻址的 FADU 中根节点的类型为 Node-Descriptor-Data-Element 的数据元素和根节点数据单元中所有类型为 File-Contents-Data-Element 的数据元素都要被传送。

7.5.6 UA——无结构的所有数据单元访问上下文

在 UA 访问上下文中,仅传送被寻址的 FADU 中所有类型为 File-Contents-Data-Element 的数据元素。

表 1 按访问上下文读的结果

访问上下文	结 果
HA	单个子树
HN	未定义的类型——因为传送 data-exists = TRUE 的 Node-Descriptor-Data-Element 时不包括构成其 DU 的数据元素。
FA	每个元素只具有一个节点的子树串
FL	每个元素只具有一个节点的子树串
FS	具有一个节点的单个子树
UA	单个 DU
US	单个 DU

7.5.7 US——无结构的单个数据单元访问上下文

在 US 访问上下文中,传送属于被寻址的 FADU 中根节点数据单元的类型为 File-Contents-Data-Element 的所有数据元素。

7.5.8 访问上下文概要

依据定义在 ISO 8571-FADU 里的数据类型,按一个访问上下文去读一个非空分级子树的结果如表 1 所示。

7.6 标识结构

FADU 通过引用相应子树根节点来标识,一个 FADU 可由下列机制中的任何一个来标识:

- first: 标识文卷结构的前序遍历序列中第一个有数据存在的 FADU;
- last: 标识文卷结构的前序遍历序列中的最后一个 FADU;
- previous: 标识文卷访问结构的前序遍历序列中当前已标识的 FADU 之前的 FADU;
- current: 在文卷中的当前位置保持不变(见第 8 章);
- next: 标识文卷访问结构的前序遍历序列中紧随当前已标识的 FADU 之后的 FADU;
- begin: begin 的确切意义取决于所使用的约束集,但是应使定位 next 能标识由 first 所确定的 FADU;
- g) end: 建立没有当前位置的文卷状态。但是使用 previous 将在文卷访问结构的前序遍历序列中标

识由 last 所确定的 FADU;

h) NodeName:指出要被标识的 FADU 的标识符。对指定的节点名的搜索被限制在当前定位节点的子节点中;

i) 节点名序列:规定从该文卷根节点到被定位节点的 FADU 标识符的路径,第一个节点名是文卷根节点的子节点,所以根节点本身是由节点名的空序列来标识;

j) 节点号:通过在文卷访问结构的前序遍历序列中的编号指出要选择的节点,文卷根节点的编号为零。

7.7 约束集

通过定义一些约束集来约束通用分级结构并进一步限定它的动态性。在使用中,每个约束集标识一个特定的结构,它们规定在附录 A 中。约束集在协议交换中按名引用,这些名的 ASN.1 类型为 OBJECT IDENTIFIER。

实际上,最终在打开时期里允许的动作集合,相对于在约束集里规定的动作,将受到进一步的约束。这种进一步的限制是通过允许的动作和访问控制等文卷属性、处理方式和并发控制等活动属性以及为当前 FTAM 时期选择的功能单元实施的。

8 对文卷的动作

虚文卷存储器定义操纵文卷中数据单元的动作。各个动作的定义(见第二篇)陈述它所适用的数据单元和对那些数据单元产生的影响。某些动作还建立文卷存储器状态,诸如状态“file selected”,或者 FADU 位置。

动作由服务原语调用,它们的语义与那些定义在文卷服务 GB/T 16505.3 中的服务原语的语义一起定义(见注 1)。

每个动作的使用受响应者访问控制的支配(见 12.16)。动作的适用性在那个动作期间和动作之后受管理并行活动的并发控制的支配(见注 2 和注 3)。

注

- 1 服务定义的某些原语与特定的动作密切相关,但是逻辑上两者总是不同的。
- 2 当同意使用托付、并发和恢复时(见 ISO 9804),应按那个标准的规定运用并发控制。
- 3 并发控制允许一个访问者为一个资源上的各种动作请求;动作不需要、共享访问、互斥访问、或者不许访问。如果请求并许可了不许访问或互斥访问,则其他联系的动作不能发生。如果请求并许可了共享访问,则其他联系的互斥访问不能发生。
- 4 响应者执行动作(就象在第二篇中定义的那样)时应使它们可串行化,可串行化的执行被定义为——使活动并发执行的效果就象同样活动串行执行的效果一样。串行执行是指每个动作在下一个动作开始之前完成。

8.1 与批量数据传送的关系

在 ASN.1 模块 ISO 8571-FADU 中定义了文卷内容的分级模型,这个模型定义了称做文卷结构项的三种数据类型,它们是:

a) Subtree:该模块的主入口点数据类型,它代表一个完整的 FADU;

b) Children:由一个 enter-subtree 项、一个或多个子树构成的序列、和一个 exit-subtree 项的序列构成的数据类型。这种数据类型在上级使用时表示的并不是单个 FADU,而是限定请求特殊动作的一系列 FADU(见 8.3);

c) DU:仅由单个数据单元组成的数据类型。

当被传送的批量数据的接收者接收到对应于一个文卷结构项的一系列具 Data-Element 数据类型的值时,就说他辨认出了该文卷结构项,即这些 Data-Element 类型对应该文卷结构项的各个数据类型。

某些约束集区别一些动作的两种变体,为它们定义不同的语义。它们被称为

d) 在上级由传送完整子树而引起的常规动作;

e) 传送一个子节点文卷结构项而引起的限定动作,此时在子节点数据值内按照每棵子树的次序运

用限定动作。

注：在通用分级约束集里，这种机制被用来区分“姐妹节点插入”和“子节点插入”动作。

8.2 读批量数据传送

在实施 FADU 读动作时(见 11.2)，响应者应根据所用的访问上下文生成一个或多个 Subtree 文卷结构项，或者一个单独的 DU 文卷结构项。

8.3 写批量数据传送

在实施写动作之前，响应者应使用所提供的 FADU 标识完成一个定位动作。

当执行插入、置换或扩展动作时(见 11.3 到 11.5)，响应者应对收到的一系列数据值作语法分析以辨认文卷结构项。可接受的文卷结构项取决于使用中的约束集，但通常响应者将会辨认到：

- a) 能够作为子树被辨认的一系列数据值；这种文卷结构项表示一个 FADU。
- b) 能够作为子节点被辨认的一系列数据值；这表示欲对其实施限定动作的一系列 FADU。在对这些文卷结构项是合法的各个约束集里规定了限定动作的意义和它的效果。
- c) 能够作为 DU 被辨认的一系列数据值；且它不可辨认作为由一系列更长的数据值组成的任何其他文卷结构项的部分；这种文卷结构项表示一个数据单元。在某些约束集中，这种文卷结构项用来扩展或置换当前被定位的 FADU 的数据单元。

注

- 1 当用 FTAM 进行传送时，数据值经由 F-DATA indication 原语传递到响应者。
- 2 Children 文卷结构项由一个 enter-subtree 项、一系列 Subtree 项和一个 exit-subtree 项组成(见 7.2)。
- 3 使用动作限定符机制，例如在通用分级文卷中用来区分插入动作；常规动作是当作姐妹节点插入，而限定动作是作为子节点插入。
- 4 DU 文卷结构项不是自定界的，它由传送数据的结束或者由后续的 Subtree 或 Children 文卷结构项来终止。
- 5 通常对文卷结构项的辨认会要求朝前多看一个数据值。

对于接收到的每个文卷结构项，响应者应：

- d) 对收到的 Subtree 或 DU，或者对在 Children 中限定系列的每个成员整体地执行传送所请求的动作。
- e) 在完成每个动作之后，按约束集里的规定修改当前位置。

如果出现差错就立即终止这个过程。若是没有发生故障，则在为被辨认的文卷结构项完成规定的动作后，响应者应试图辨认下一个文卷结构项并重复上述过程。该过程一直继续，直到数据传送终止或者检测到差错时为止。接收到无法被辨认为一个完整文卷结构项的一系列数据值是一种差错。然而，一次数据传送中可以包含多个文卷结构项。

9 属性

9.1 属性的作用域

属性分为两种类别：

- a) 文卷属性：每个文卷由一组文卷属性值来描述。文卷属性的作用域是虚文卷存储器，且当由一个发起者的动作导致文卷属性值改变时，此新值对此后读那个属性的任何其它发起者都是可见的。
- b) 活动属性：每个活动发生在一个 FTAM 时期之内并由一组活动属性值来描述。活动属性的作用域至多是那个 FTAM 时期，且每个 FTAM 时期有一组不同的和独立的活动属性值集合。有两类性质不同的活动属性。

- 1) 现役属性与文卷属性一一对应。

注：大多数情况下映射是无足轻重的，因为许多文卷属性在文卷创建时候就被固定了。然而有几种现役属性具有不同的值，诸如现役内容类型和现役法律限定等，此值是对应文卷属性值的子集。

- 2) 当前属性涉及发起者，通常是从协议交换的参数导出。

注：当前属性并不严格地等价于静态文卷属性，但在某些情况下是密切相关的。例如当前访问口令必定是访问控制

属性里的访问口令项中的成员。

9.2 标量、向量和集合属性

属性分为三种类型：

- a) 标量属性：在给定的时刻每个标量属性有一个值；
- b) 向量属性：每个向量属性的值是零个元素或多个元素的表，其中每个元素具有一个独立的值，元素按顺序排列。
- c) 集合属性：每个集合属性的值本身是零个元素或多个元素的无序集合，集合中的每个元素具有不同值。

对每一个不同类型属性，改属性动作的效果在 10.3 中定义。

9.3 属性值

对于每个属性，定义它的值类型（对于向量或集合属性而言，它的元素的值类型）是下列之一种：

- a) 来自 GB/T 16262(ASN.1)中由 GraphicString 定义的字符集的字符序列（不包括空格字符）；
- b) 由 GB/T 16262(ASN.1)中 OCTET STRING 定义的八位位组序列；
- c) 由 GB/T 16262(ASN.1)中 INTEGER 定义的整型；
- d) 由 GB/T 16262(ASN.1)中 BOOLEAN 定义的布尔型；
- e) 在 ISO 8650(ACSE)内的 ASN.1 模块 ACSE-1 里由 AE-title 定义的应用实体标题；
- f) 日期和时间；日期和时间值是被限于定义在 GB/T 7408 中的可用 GeneralizedTime 表示的那些值（见注 1）；
- g) 在 GB/T 16505 这部分由枚举定义的有名值集中的一项；
- h) 定义在 GB/T 16262 里的 OBJECT IDENTIFIER；
- i) 不能够确定该值的指示（见注 2）。

注

- 1 日期和时间所具有的精确度和清晰度由响应者确定和维护，不在 GB/T 16505 这部分定义。可选地，使用表示的右截断以指示精度（见 GB/T 7408）；而不采用左截断。
- 2 例如，当该实系统环境已不能保存该信息时，应该作出这种响应。

9.4 文卷属性的支持

在文卷存储器里为文卷属性定义了三级支持：

- a) 不予支持的属性；发起者决不引用它不支持的属性；试图引用响应者不支持的属性会导致差错。
- b) 部分支持的属性；发起者不应宣称对任何属性的部分支持，发起者若引用被响应者部分支持的属性将得到“no value available”的结果；改变这种属性的试图将不会成功。
- c) 支持的属性；宣称支持某属性的发起者在请求文卷存储器动作时应至少能在该属性的最小值域内指定属性值，这在第十五章中定义。

注：这种能力可服从于一致性测试。

引用由响应者支持的属性将产生有意义的值，该值至少属于如第 15 章定义的那种最小值域。宣称支持某属性的响应者应支持定义在第 12 和 13 章中的与该属性相关联的语义。

第二篇 文卷存储器上的动作

10 对整个文卷的动作

10.1 创建文卷

该动作创建新文卷或者可选地选择一个已存在的文卷，并建立该新文卷的属性。它建立文卷选择时期并选择这个新创建的文卷（见 10.2）。该文卷的初始状态由它的约束集中的创建状态项定义（见附录 A）。

10.2 选择文卷

该动作在发起者和一个特定文卷之间建立关系。此动作建立文卷选择时期,它是 10.3 到 10.8 中的动作的先决条件。为了成功地选择文卷,参数应精确地指定一个文卷。

10.3 改属性

该动作改变现存的文卷属性。

- a) 对于标量属性,该动作置换该属性的现存值;
- b) 对于向量属性,该动作给出的表置换整个的元素表;
- c) 对于集合属性,该动作:
 - 1) 在该属性上添加给定的一个或多个元素;
 - 和/或
 - 2) 从属性里移去等于给定值的一个或多个元素。

注

- 1 成功地添加一个元素到集合属性,要求被增添的那个元素不同于那个集合里的所有元素。
- 2 成功地从集合属性里移走一个元素,要求在那个集合里存在该元素。

10.4 读属性

该动作询问所请求的属性值。对于向量或集合属性,它返回完整的元素值表。

10.5 打开文卷

该动作建立文卷打开时期,以便在被选择的文卷上实施文卷访问动作(见第 11 章)。当文卷为读而打开时,仅读和定位动作有效;当文卷为写而打开时,所有文卷访问动作都有效。

10.6 关闭文卷

按照有序方式,该动作终止从前被打开文卷动作所建立的打开时期。

10.7 删除文卷

该动作删除并去选已选择的文卷,它终止当前文卷选择时期。

10.8 去选文卷

按有序方式,该动作终止当前文卷选择时期。

11 文卷访问动作

访问动作在由打开文卷动作建立起的时期内操作。在具体文卷上可用的动作将取决于适用于该文卷的约束集并可被该约束集修改(见附录 A)。此外文卷属性“允许的动作”和“访问控制”会进一步约束某些动作是否可在这个文卷上实施。

FADU 标识方式的可用范围可由约束集修改。供访问的文卷内容结构是在第 7 章中定义的文卷访问结构,它在执行读动作时由使用中的访问上下文修改。

11.1 定位

该动作定位指定的 FADU。参数值包括请求在文卷的起始或结尾的位置定位。文卷打开时的位置按所使用的约束集中的规定。

允许通过参数值指出 FADU 标识的不同方式。在 7.6 里定义了合法的 FADU 标识值的范围。

如果定位操作失败,当前位置保持不变。

11.2 读

该动作定位并读一个 FADU。访问之后的位置不改变。

基于所要求的访问上下文决定待传送的数据单元和结构信息。请求可以针对规定在约束集里的任何已定义的访问上下文。

11.3 插入

该动作创建一个新的 FADU 并按约束集里的规定把它插在文卷中的某个位置。(见附录 A 和 D)。

在分级文卷里 FADU 可插在相对于父节点的任一级,此级被规定在 FADU 的 Node-Descriptor-Data-Element 之中(见 7.2)。缺省的相对级是 1。

11.4 置换

该动作置换现存的 DU 或 FADU 的内容,从前的内容将丢失。置换或者针对当前定位的整个 FADU,或者只针对连接到它根节点的 DU 之内容。

a) 仅当文卷结构信息在置换 FADU 时是可用的,置换整个 FADU 才合法。置换动作既不能改变当前定位的 FADU 节点名,也不能改变它的根节点的弧长。置换对 FADU 里的结构信息和 DU 都适用,也即新结构信息可能被引进当作置换的一部分。

b) 仅当被置换的 DU 不影响文卷结构信息时,置换连接到当前定位的 FADU 根节点的 DU 才被允许。

置换动作不改变当前的位置。

11.5 扩展

该动作把数据加到与当前定位的 FADU 根节点相关联的 DU 之末端。扩展动作仅适用于现存的 DU,扩展动作并不改变当前位置。

11.6 擦除

擦去一个 FADU,并按前序遍历序列定位在该 FADU 之后剩余的 FADU 中的第一个。当定位在文卷的根节点时执行擦除动作,那么该文卷恢复到刚创建时的状态。

11.7 文卷动作与当前位置

表 2 定义了文卷访问动作对当前位置的影响。如果调用动作的服务原语携带 FADU 标识符,那么当前位置可能已经被一个蕴含的定位所建立。如果定位动作不成功,则当前位置不改变。

注:附录 D 包含了动作对当前位置影响的例子和建立结构的途径的例子。

表 2 动作对位置的影响

访问操作	访问后的位置
定位	如 FADU 标识所规定的那样,定义在 11.1
读	由 FADU 标识所规定
插入	见附录 A
置换	不改变
扩展	不改变
擦除	跟在擦除的 FADU 之后的 FADU(按前序遍历序列)

第三篇 属性定义

12 文卷属性

每个文卷属性都是全局的,在任一特定时刻,在全局里它有一个值或一组值。文卷动作的所有发起者对一个文卷属性都将看到同样的值、值集或者都得到“no value available”指示。

12.1 文卷名

在文卷存储器里的每个文卷都有一个文卷名。文卷名是向量属性,它们中的元素形成名字组成部分的序列,每个组成部分都是 GraphicString 类型的值(见第 15 章)。

文卷名属性值在文卷创建时设置,而且能够由改属性动作修改。

注

1 GB/T 16505 并不为文卷名的各组成部分定义任何解释;这为发起者和响应者提供了透明的命名机制。

2 名在虚文卷存储器中定义的组成部分和它在实系统环境中划分的组成部分之间的关系由本地实现选择。一个实现可把本地组成部分结构映射到文卷名属性的组成部分上,或者它可以把现存的文卷名语法映射到从虚文卷存储器看来只有一个组成部分的文卷名上。

- 3 一个实现可以将选择到该文卷的访问途径反映到虚文卷存储器文卷名的组成部分,但是这种选择本身对互连目的并不是可见的。

12.2 允许的动作

允许的动作属性是向量属性,它指明能够在文卷上执行的动作集合,和能够被用于文卷访问动作的 FADU 标识方式的集合(见 7.6 和第 11 章)。响应者可用任何能够将其映射到所用实系统的方法来实现这个允许的动作集。

FADU 标识的不同方式被分成如下三种 FADU 标识组:

遍历:begin,first,next,last,end;

反向遍历:begin,first,previous,last,end;

随机次序:Current,单个 NodeName,节点名序列,节点号。

该属性是向量,其元素取布尔值,每个元素指明一个动作或 FADU 标识组的可用性,这些元素是:

a) 可用的动作

- 1) 读
- 2) 插入
- 3) 置换
- 4) 扩展
- 5) 擦除
- 6) 读属性
- 7) 改属性
- 8) 删除文卷

b) 可用的 FADU 标识组

- 1) 遍历
- 2) 反向遍历
- 3) 随机次序

该属性的值在文卷创建时设置,且不能用改属性动作加以改变。

12.3 内容类型

内容类型属性指明文卷内容和结构信息的抽象数据类型。如果在传送文卷期间必须维护完整的文卷结构和语义,则它是必需的。内容类型是标量属性。

该属性的值在文卷创建时设置,且不能用改属性动作改变。

该值或者是文件类型名,可能带有任何类型的单值参数;或者是抽象语法名和约束集名对。这些名字都是类型 OBJECT IDENTIFIER 的值。

如果用于写的打开动作给出的内容类型信息不与这个属性相匹配,那么这个打开动作就不会成功。

如果用于读的打开动作给出任何等于、简化或松弛于这个属性(见附录 B)的信息,那么这个打开动作会成功。更进一步说,给出的信息可以是:

- a) 不提供,在此情况使用该属性的值;
- b) 抽象语法名和约束集名,在此情况匹配必须准确;
- c) 没有参数的文件类型名,在此情况对于写必须准确匹配,而对于读可以是已定义的简化;
- d) 带有空参数的文件类型名,在此情况名字必须匹配,或对读是简化,但参数值是从该属性值中提供;
- e) 带有非空参数的文卷类型名,在此情况名字必须匹配,或对读是简化,并且对于写,参数必须匹配或者是更受限,而对于读,则要求参数匹配或者是松弛。

12.4 存储器帐户

存储器帐户属性标识对累计的文卷存储器费用负责的可结算机构,它是标量属性。

该属性的值在文卷创建时设置,但能用改属性动作来修改。

存储器帐户属性的值是 GraphicString 类型。

注:管理帐户的机制和协商收费率的手段不属 GB/T 16505 这部分的范围。

12.5 创建日期和时间

创建日期和时间属性指明该文卷创建于何时,它是标量属性。

当文卷被创建时由响应者设置它并且引用响应者的本地日期和时间。日期和时间属性值是创建动作执行时的日期和时间。

不能够使用改属性动作来修改该属性。

该属性值的类型是 GeneralizedTime(见 9.3)。

12.6 上次修改的日期和时间

上次修改的日期和时间属性是标量属性,它指示文卷的内容上次是何时被修改的。

无论何时只要响应者为修改或扩展而打开文卷后又关闭了它(包括一个连接失败后的关闭),就改变该属性。除非被打开的文卷允许改变内容,否则该属性不改变;文卷属性改变时,并不改变它。该属性值为执行文卷关闭动作时的日期和时间。

不能使用改属性动作来改变此属性。

属性的值是 GeneralizedTime 类型(见 9.3),对于新创建的文卷,该值等于创建日期和时间属性的值。

12.7 上次读访问的日期和时间

上次读访问的日期和时间是标量属性,它指明上次何时文卷的内容被读过。

无论何时只要响应者为读而打开文卷后又关闭了它(包括一个连接失败后的关闭),就改变该属性。该属性值为执行文卷关闭动作时的日期和时间。除非文卷被打开,否则该属性不改变;如果文卷仅被选择但没被打开,例如读它的属性,则该属性并不改变。

不能够使用改属性动作来改变此属性。

该属性的值是 GeneralizedTime 类型(见 9.3),对于新创建的文卷,该值等于创建日期和时间属性的值。

12.8 上次属性修改的日期和时间

上次属性修改的日期和时间属性是标量属性,它指明文卷属性值上次何时被修改。

当对一个或多个属性成功地执行了改属性动作后,响应者改变该属性。一个属性(如文卷大小)的隐含变化并不改变该属性。该属性值为执行改属性动作时的日期和时间。

该属性不能由改属性动作来改变。

该属性的值是 GeneralizedTime 类型(见 9.3),对于新创建的文卷,该值等于创建日期和时间属性的值。

12.9 创建者标识

创建者标识属性是标量属性,当文卷被创建时由响应者设置它。该标识被设置成文卷创建时当前发起者标识活动属性的值。

不能够使用改属性动作改变该属性。

该属性值是 GraphicString 类型。

12.10 上次修改者标识

上次修改者标识属性是标量属性。无论何时只要响应者为修改或扩展而打开文卷后又关闭了它(包括一个连接失败后的关闭),就改变该属性。当文卷关闭时,该标识被设置成当前发起者标识活动属性的值。

不能够使用改属性动作改变该属性。

该属性值是 GraphicString 类型。对于新创建的文卷,此值等于创建者标识属性的值。

12.11 上次阅读者标识

上次阅读者标识属性是标量属性。无论何时只要响应者为读而打开文卷后又关闭了它(包括一个连接失败后的关闭),就改变该属性。该标识被设置成文卷关闭时当前发起者标识活动属性的值。

不能够使用改属性动作改变该属性。

该属性值是 GraphicString 类型。对于新创建的文卷,此值等于创建者标识属性的值。

12.12 上次属性修改者的标识

上次属性修改者标识属性是标量属性。当对一个或多个属性成功地执行了改属性动作后,由响应者改变它。该标识被设置成当前发起者标识的值。

不能够使用改属性动作改变该属性。

该属性值是 GraphicString 类型。对于新创建的文卷,此值等于创建者标识属性的值。

12.13 文卷可用性

文卷可用性属性是标量属性,它指明在该文卷能打开之前是否要期待延迟。

该属性的值在文卷创建时设置,但能由改属性动作改变。

其值可以是“immediate availability”或“deferred availability”。

注:此属性指明了对响应者有意义的一种文卷分类,而不是延时量的度量。如果此值起初设置成“immediate availability”则该文卷被假定保存在非可拆卸装置上,并且当访问此文卷时将不会遇到明显的延迟。如果发起者设置该值为“deferred availability”,则它可能是给响应者的一个指示,即此文卷可存储在可拆卸装置上。如果发起者将该属性重置为“immediate availability”,则它可以是向响应者指明在此文卷已经被下载到可拆卸装置的情况下,拷贝此文卷到立即可用的装置上。然而该属性的实际使用是依赖于实现的。

12.14 文卷大小

文卷大小属性是标量属性。无论何时只要响应者为修改或扩展而打开文卷后又关闭了它(包括一个连接失败后的关闭),就改变该属性。在文卷关闭时该属性值被设置成整个文卷按八位位组表示的标称值。(见注2)

该属性不能使用改属性动作来改变,或用创建动作的初始属性参数来设置。

该属性值是整型。对新创建的文卷,此值被置为零。

注

- 1 文卷大小基于实文卷存储器所涉及的数据类型的具体表示。当数据按不同形式移动或存储时,取决于协商的传送语法,文卷的大小可变化。
- 2 如果实文卷存储器给文卷分配空间是按基本单位(如块、存储桶)的倍数计,则也许不能确定文卷中确切的八位位组数目,那么就只能假定文卷大小是按基本单位的倍数计(按八位位组)。

12.15 未来文卷大小

未来文卷大小属性是标量属性,它指明作为修改和扩展的结果,此文卷可能扩展到按八位位组表示的标称值。

该属性的值在文卷创建时设置,且能够用改属性动作修改。

该属性值是整型数。

注:在当前文卷大小属性值达到未来文卷大小属性值时,响应者可以:

- a) 增大未来文卷大小;
- b) 增大未来文卷大小并给出警告;
- c) 不增大未来文卷大小,但指明差错。

12.16 访问控制

访问控制属性是集合属性,它定义文卷合法访问的条件。

集合中的每个元素给出一个文卷合法访问的条件,至少要满足这些条件中的一个,才能允许访问该文卷。然而,访问必须基于一个单个条件,而不是基于多个独立的条件的并集。

该属性的值在文卷创建时设置,且能够用改属性动作来改变。

一个条件包括一或二项,说明它所允许的访问(见下面 a)),并且还包括给出基于各种活动属性值测试的零到三个其它项(见下面 b))。只有其中各项都满足时则整个条件才被满足。

这些项具体是:

a) 访问的说明,包括

- 1) 一个动作表项,按照布尔向量的形式;如果某动作存在于该表内则对应的布尔量为真,各布尔量分别对应于动作读、插入、置换、扩展、擦除、读属性、改属性和删除文卷。

每当要设置当前访问请求属性时,就使用此动作表项。仅当包含在当前访问请求属性值里的动作全部出现在动作表中,该动作表项才被满足。

如果该动作表项被满足,那么选择文卷和去选文卷动作也是允许的。如果被满足的该动作表项包含任何一种文卷访问动作(见第 11 章),则动作打开文卷和关闭文卷也是允许的。只要动作定位在建立的 FTAM 时期里是可用的,则它也是允许的。

- 2) 可选地,按照并发键向量的形式出现的并发访问项;并发键分别对应于动作读、插入、置换、扩展、擦除、读属性、改属性和删除文卷。并发键是布尔向量,它的元素对应于并发锁的可能值(见 13.9);在并发键里的元素对应于锁值“not required”,“shared”,“exclusive”,和“no access”。

每当要设置当前并发控制属性时,就使用该属性。仅当在并发键里对应于当前并发控制属性中所请求的锁的布尔量为真时,该并发控制项才被满足。

如果从访问控制元素中省略了该项,那么并发控制参数不应存在于由这种访问控制元素允许的访问之中。

b) 测试项,包括

- 1) 可选地,一个标识;此值是 GraphicString 类型。如果它与联系的当前发起者标识动作属性值(见 13.3)相匹配,则该项被满足。
- 2) 可选地,按照串(可有一个或多个是空白)向量形式出现的口令;向量的每个元素对应于动作读、插入、置换、扩展、擦除、读属性、改属性和删除文卷之一。如果该向量中的每个非空串都与保持在当前访问口令属性里的对应口令相匹配,则该项被满足。
- 3) 可选地,一个位置;其值是一个应用实体标题。如果它与 FTAM 时期的当前主呼应用实体标题属性的值(见 13.6)相匹配,则该项被满足。

注

- 1 当文卷生成时,访问控制属性的初始值被分配。如果当文卷生成时访问控制属性不被支持,则它由一个局部响应者事件决定这个值的分配。
- 2 该属性可被一次只允许一个访问控制条件(长度为一的表)的实现所支持。
- 3 某些实现可能对条件中各项可能的组合方式,或者条件的各种可能存在形式之数目加以限制。例如,一个实现可能只允许每个被容许的访问类型只有一个口令。
- 4 标识、口令或应用实体标题的匹配并非必须是基于文字上的相同。例如,个体名字可匹配组名,只要此个体是那个组的成员。

12.17 法律限定

法律限定属性是标量属性,它运送关于文卷的法律状态和它的使用的信息。

该属性值在文卷创建时设置,它能够被改属性动作改变。

该属性值是 GraphicString 类型。

注:支持这个属性的含义支持将取决于国家数据保护立法的形式。

12.18 专用

专用属性是标量属性,它的含义不定义在 GB/T 16505 这部分。

该属性值可在文件创建时设置。它可由改属性动作来改变。

该属性值能够取任何一种形式。

注：非常不鼓励使用这个属性，提供它只是为了在当没有本标准内部属性的组合能够运送预期的语义的情况下使用。

13 活动属性

活动属性被用作描述性工具，以反映进展中的 FTAM 时期的状态。对于每个初始化了的 FTAM 时期存在有一组活动属性值。这些属性值只反映文卷存储器访问的状态，而并不反映到文卷存储器通信路径的详细状态。

13.1 现役内容类型

现役内容类型属性是标量属性，它或者标识文件类型名，或者标识为文卷传送而建立的一对抽象语法名和约束集名。该属性值在此文卷打开时建立，可从内容类型属性中得出或从发起者提供的取代值中得出。

该值或者是文件类型名，它可能带有任何类型的单值参数；或者是抽象语法名和约束集名对。这些名字都是类型 OBJECT IDENTIFIER 的值。

现役内容类型属性的作用域是打开时期。

13.2 当前访问请求

当前访问请求属性是向量属性，它的元素取布尔值。对于每个动作有一个元素，这些动作在一个具体的选择时期可能允许也可能不允许，因此要在 FTAM 时期中请求。

对应文卷访问动作的元素是：

- a) 读，允许文卷访问动作读；
- b) 插入，允许文卷访问动作插入；
- c) 置换，允许文卷访问动作置换；
- d) 扩展，允许文卷访问动作扩展；
- e) 擦除，允许文卷访问动作擦除；

如果任何文件访问动作被允许，则动作打开文卷和关闭文卷也是允许的。如果在建立的 FTAM 时期中定位动作是可用的，那么动作定位也是允许的。

对应整个文卷动作的元素是：

- f) 读属性，允许文卷动作读属性；
- g) 改属性，允许文卷动作改属性；
- h) 删除文卷，允许文卷动作删除文卷。

如果任何文卷动作被允许，则选择文卷和去选文卷动作也是允许的。

当前访问请求属性的作用域是选择时期。

13.3 当前发起者标识

当前发起者标识属性是标量属性，当建立 FTAM 时期时，它对响应者标识发起者。

该属性的值是 GraphicString 类型。

当前发起者标识属性的作用域是 FTAM 时期。

13.4 当前位置

当前位置属性指明在文卷内部的当前位置，是标量属性。该属性的值可被打开时期内部的任何动作（定位，擦除，读，插入）所修改。当前位置的初值在初始化时由所使用的约束集来定义（见附录 A）。

该值可以是个整数，它标识在前序遍历序列中的第 n 个节点；也可以是个 FADU 名表，它标识到某个节点的路径（见 7.6）。

当前位置属性的作用域是打开时期。

13.5 当前处理模式

当前处理模式属性是向量属性，它的元素取布尔值。在一个特定打开时期中允许的每个文卷访问动

作都有一个元素。

这些元素是：

- a) 读,允许文卷访问动作读;
- b) 插入,允许文卷访问动作插入;
- c) 置换,允许文卷访问动作置换;
- d) 扩展,允许文卷访问动作扩展;
- e) 擦除,允许文卷访问动作擦除。

如果文卷访问动作定位在建立的 FTAM 时期是可用的,则它总是合法的。

当前处理方式属性的作用域是打开时期。

13.6 当前主呼应用实体标题

当前主呼应用实体标题属性是标量属性,它指明建立该 FTAM 时期的应用实体的应用实体标题。

该属性的值是一个应用实体标题。

当前主呼应用实体标题属性的作用域是 FTAM 时期。

13.7 当前响应应用实体标题

当前响应应用实体标题属性是标量属性,它指明对应虚文卷存储器的应用实体的应用实体标题。

该属性的值是一个应用实体标题。

当前响应应用实体标题属性的作用域是 FTAM 时期。

13.8 当前帐户

当前帐户属性是标量属性。它标识在 FTAM 时期与文卷使用有关的,响应者可向其征收费用的可结算机构。

该属性的值属于 GraphicString 类型。

当前帐户属性的作用域是 FTAM 时期,然而在选择时期可用一个替换值来累计对一个特定文卷的各动作的费用,对这种情况,该替换值在文卷选择时建立,并在该文卷去选时恢复到原属性值。

注:该属性并不一定要指出由文卷服务提供者征收的有关通信费用的责任。

13.9 当前并发控制

当前并发控制属性指明关于发起者所请求的并行访问方面的限制,这些所要求的并行访问动作都定义在 13.2 中。

当前并发控制属性是向量属性,它的元素对应于列于 13.2 内的各访问动作。向量的每个元素的值为表 3 给出的四种值之一。

当前并发控制属性的作用域不是选择时期就是打开时期,这取决于它的值是由选择文卷动作建立还是由打开文卷动作建立。

表 3 并发控制选项

	我可以执行此动作	其他人可以执行此动作
Not Required	不可以	可以
Shared	可以	可以
Exclusive	可以	不可以
No Access	不可以	不可以

13.10 当前锁定方式

当前锁定方式属性是标量属性,它指明是否 FADU 锁定机制已被使能。在文卷打开时从使能 FADU 锁定参数得出此值。

该属性的值是布尔量;如果 FADU 锁定被使能时,则它的值为真。

当前锁定方式属性的作用域是打开时期。

13.11 当前访问口令

当前访问口令属性是向量属性。每个元素给出一个同当前访问请求属性的八个元素之一相联系的值。用选择或者创建文卷服务原语的参数来设置这些值。

当文卷选择时期被释放时,这些值被复位。按访问控制属性(见 12.16)的值来验证这些值以确定访问是否可以。

该属性每个元素的值是 GraphicString 类型或者是 OCTET STRING 类型。

当前访问口令属性的作用域是选择时期。

注:此属性可被用来运送口令,或能力,或某公开密钥加密模式的元素。

13.12 现役法律限定

现役法律限定是标量属性,它的意义和它与文卷属性法律限定的关系都不定义在 GB/T 16505 里。当文卷被打开时建立此属性值,且从法律限定属性中得出。

该属性的值是 GraphicString 类型。

现役法律限定属性的作用域是打开时期。

14 属性组

GB/T 16505 这部分大范围地规定了一个实现可支持的属性(见 9.4)。为了指定支持的级别,这些属性被置于如下规定的组里,对于每个组或者

- a) 对整组都没有支持;或者
- b) 支持或者部分地支持组内每个属性,并当被询问时,或者提供一个有意义的响应或者提供一个“no value available”指示。

注:在 FTAM 协议(GB/T 16505.4)里,可用的属性组在 FTAM 时期建立时指明。

14.1 核心组

下列的属性总被定义,且在任何引用虚文卷存储器的规范里是可用的。对于这些属性,不能够提供值“no value available”:

- a) 文卷属性
 - 1) 文卷名
 - 2) 允许的动作
 - 3) 内容类型
- b) 活动属性
 - 1) 现役内容类型
 - 2) 当前访问请求
 - 3) 当前发起者标识
 - 4) 当前位置
 - 5) 当前处理模式
 - 6) 当前主呼应用实体标题
 - 7) 当前响应应用实体标题

14.2 存储组

如果对于全部支持的属性,响应者保证信息的存储,并允许一个活动引用以前某个活动建立的信息,那么下列的属性才是有意义的。这些属性中的任何一个都可取值“no value available”。指定存储组则允许引用下列定义:

- a) 文卷属性
 - 1) 存储器帐户
 - 2) 创建的日期和时间
 - 3) 上次修改的日期和时间

- 4) 上次读访问的日期和时间
- 5) 上次属性修改的日期和时间
- 6) 创建者标识
- 7) 上次修改者标识
- 8) 上次阅读者标识
- 9) 上次属性修改者标识
- 10) 文卷可用性
- 11) 文卷大小
- 12) 未来文卷大小
- b) 活动属性
 - 1) 当前帐户
 - 2) 当前并发控制
 - 3) 当前锁定方式

14.3 安全组

下列属性同安全性和访问控制有关,它们构成安全组,这些属性中的任何一个可取值“no value available”。

- a) 文卷属性
 - 1) 访问控制
 - 2) 法律限定
- b) 活动属性
 - 1) 当前访问口令
 - 2) 现役法律限定

14.4 专用组

下列属性是在 OSI 标准范围之外的,它是独立的,并仅当专用组被规定时才可规定。

- a) 文卷属性
 - 1) 专用

15 最小属性值域

本章定义了一个属性值域的有限集合称作“虚文卷存储器最小属性值域”。它们是支持一个属性时交互工作的最小要求(见注)。它们适用于定义在第 12 章和第 13 章中的,具有通用数据类型如整型或字符串型的属性。对于属性的值是枚举型或者是枚举的向量,即使只提供部分的支持,也应能接收或产生枚举值,而不给出差错指示。

对于直接对应于文卷属性的活动属性,适用于该对应文卷属性的最小值域应同样适用于它。

除非显式地声明,否则字符串只限于由 ASN.1 GraphicString 所定义的打印字符,且不应包括空格字符。至少作为字符集登记项 2 登记的 G0 字符集应被支持,不要求支持任何转义序列。

所有的值域都是闭区间。

注

- 1 这些最小属性值域可使引用 GB/T 16505 这部分的其它标准规定最小值集,其要求任何开放性实现——(发起者或者响应者)可以发送或接收。它们还必须规定对于哪些属性只支持“no value available”值。
- 2 虚文卷存储器最小属性值域可能对于许多特定文卷服务的使用来说是限制太多了。这些值域的定义并不以任何方式约束具体的实现发送和接收在已定义的值域之外的适当类型的任意属性值。
- 3 在表 4 中规定的活动属性的最小值域实际上限制了用于对应服务原语(见 GB/T 16505.3)中参数的值域。

表 4 活动属性

属 性	类 型	最小属性值域
当前访问请求	布尔向量	见注
当前发起者标识	GraphicString	1~8 字符
当前位置	Node-Descriptor-Data-Element	见注
当前处理模式	布尔向量	见注
当前主呼应用实体标题	应用实体标题	无最小要求
当前响应应用实体标题	应用实体标题	无最小要求
当前帐户	GraphicString	1~8 字符
当前并发控制	枚举的向量	见注
当前锁定方式	布尔	无最小要求
当前访问口令	GraphicString 或 OCTET STRING	0~8 字符或八位位组

注：不能够为这种活动属性规定最小值域，这种属性可取的值取决于发起者和响应者的能力并且可能受到使用中的约束集和/或允许的动作文卷属性和/或访问控制文卷属性等的进一步限制。

表 5 文卷属性

属 性	类 型	最小属性值域
文卷名	GraphicString 的向量	单个元素
允许的动作	每个元素	1~8 用字母开头的大写字母或数字
内容类型	布尔向量	任何已定义值的非空子集
存储器帐户	抽象语法, 约束集名对或者文件类型名	无最小的要求
创建日期和时间	GraphicString	1~8 字符
上次修改日期和时间	日期和时间	按 GB/T 7408
上次读访问日期和时间	日期和时间	按 GB/T 7408
上次属性修改日期和时间	日期和时间	按 GB/T 7408
创建者标识	日期和时间	按 GB/T 7408
上次修改者标识	GraphicString	1~8 字符
上次阅读者标识	GraphicString	1~8 字符
上次属性修改者标识	GraphicString	1~8 字符
文卷可用性	GraphicString	1~8 字符
文卷大小	枚举	任何已定义值的非空子集
未来文卷大小	整型	无最小的要求
访问控制	整型	无最小的要求
动作表	条件的集合	单个条件
并发访问	布尔向量	任何已定义值的非空子集
标识	布尔向量的向量	任何已定义值的非空子集
口令	GraphicString	1~8 字符
位置	GraphicString 或 OCTET STRING	0~8 字符或八位位组
法律限定	应用实体标题	所有应用实体标题
专用	GraphicString	0~80 个 GraphicString 和/或空格字符
	任意	不适用

附 录 A
(标准的附录)
文卷访问结构约束集

A1 一般考虑**A1.1 约束集功能**

第7章定义了通用分级结构,而第8章和第11章定义了对其可适用的动作。通过定义一些约束集来约束这种通用结构并进一步限定它的动态性。每个约束集标识一种常用的具体结构。约束集在协议交换中是借助其值为ASN.1类型OBJECT IDENTIFIER的名来引用的。

相对于约束集里规定的动作,文卷打开时期最终允许的动作集实际上还受到进一步的约束。这种进一步约束由允许的动作文卷属性、处理模式活动属性和为当前FTAM时期选择的功能单元所施加。

除了那些在约束集里的规定之外,在文卷内容上附加的约束还能够通过文件类型中的规范来施加(例如文件类型可能进一步限制可用的数据类型)。

A1.2 与批量数据传送的关系

通过重复地应用被描述的动作,多个FADU可以作为单个批量数据传送的结果而被插入、置换或扩展(见第8章和附录D)。

A1.3 约束集内容

约束集的定义由下列构成:

- a) 预期适用范围声明;
- b) 供在文本中引用的约束集描述符;
- c) 供在FTAM协议中引用的约束集标识符;
- d) 对节点名的约束;
- e) 有效文卷访问动作集;
- f) 在写文卷时的一些限定动作(见第8章);
- g) 可用于读文卷的文卷访问上下文集合;
- h) 文卷创建时的状态;
- i) 文卷打开时定位的FADU;
- j) 对文卷的开头的定义;
- k) 对文卷的结尾的定义;
- l) 读整个文卷的方式;
- m) 写整个文卷的方式;
- n) 关于如何修改结构的一般约束;
- o) 超过或覆盖在第11章中给出的特定文卷访问动作的专门语义;
- p) 每个文卷访问动作可允许的FADU标识形式。

A1.4 表记法

在后面的表里,下列的缩写词指明当使用特定文卷存储器动作时FADU标识的有效范围:

- a) Valid:此动作可适用到任何FADU上;
- b) leaf:如果某动作针对的FADU是叶节点,则该动作是合法的;
- c) whole:此动作针对整个文卷(也就是在根节点开始)。

A2 约束集定义

A2.1 无结构约束集

A2.1.1 适用范围

无结构约束集适用于要当作整体来传送或访问的文卷、或者被扩展的文卷,但对文卷一部分的访问是不可用的。

A2.1.2 基本约束

无结构约束集中的基本约束在表 6 给出。

A2.1.3 结构约束

无结构约束集中的所有动作均产生一个带有数据单元(虽然这个数据单元可能是空)的根节点结构。

A2.1.4 动作约束

擦除:擦除动作产生带有空数据单元的根节点。

A2.1.5 标识约束

在无结构约束集里,FADU 标识总是 first,这种形式的 FADU 标识用于所有被允许的文卷动作。

表 6 在无结构约束集中的基本约束

约束集描述符	无结构的
约束集标识符	{iso standard 8571 constraint-set(4) unstructured(1)}
节点名	没有
文卷访问动作	读,置换,扩展,擦除
限定的动作	没有
可用访问上下文	UA
创建状态	带有空数据单元的根节点
打开后位置	First
文卷的开头	不适用
文卷的结尾	不适用
读整个文卷	在具有 FADU 标识 first 的访问上下文 UA 中读
写整个文卷	用 FADU 标识 first 和文卷访问动作置换来传送单个数据单元(没有节点描述符)

A2.2 顺序平坦约束集

A2.2.1 适用范围

顺序平坦约束集适用于由 FADU 序列构造的文卷,并且可根据单个 FADU 在序列中的位置访问该文卷。

A2.2.2 基本约束

顺序平坦约束集中的基本约束在表 7 中给出。

A2.2.3 结构约束

根节点不应带有数据单元,根节点的所有子节点应是叶节点并带有数据单元;来自根节点的所有弧长应是 1。

A2.2.4 动作约束

插入:只允许在文卷结尾使用带 FADU 标识 end 的插入动作;被插入的新节点位于文卷中的现有节点之后,插入后的位置是 end。

擦除:只允许在根节点使用带 FADU 标识 begin 的擦除动作以清除文卷,结果是不带数据单元的

孤立的根节点。

A2.2.5 标识约束

同文卷动作关联的 FADU 标识应是 begin, end, first, last, current, next, previous 或者大于或等于 1 的遍历序号中的一个。能够同这些标识一道使用的动作在表 8 中给出。

表 7 顺序平坦约束集中的基本约束

约束集描述	顺序平坦
约束集标识符	{iso standard 8571 constraint-set(4) sequential-flat(2)}
节点名	没有
文卷访问动作	定位, 读, 插入, 擦除
限定的动作	没有
可用的访问上下文	FA, UA
创建状态	不带数据单元的根节点
打开后位置	根节点
文卷的开头	根节点
文卷的结尾	无被选的节点; previous 给出在遍历序列里的最后一个节点, current 和 next 给出差错信息
读整个文卷	在具有 FADU 标识 begin 的访问上下文 FA 或 UA 中读
写整个文卷	传送一连串的叶 FADU, 它们由在访问上下文 FA 里读整个文卷而产生; 用 FADU 标识 end 和文卷访问动作插入来执行传送

表 8 顺序平坦约束集中的标识约束

动作	Begin	End	First	Last	Current	Next	Previous	结点数
定位	valid	valid	valid	valid	valid	valid	valid	valid
读	whole		leaf	leaf	leaf	leaf	leaf	leaf
插入		leaf						
擦除	whole							

A2.3 有序平坦约束集

A2.3.1 适用范围

有序平坦约束集适用于由有名 FADU 序列构造的文卷, 而且可根据各 FADU 的名字作出对它们的访问, 这些名字并不必是唯一的。

A2.3.2 基本约束

有序平坦约束集中的基本约束在表 9 中给出。

注: 在写整个文卷项时, 合并指将被传送的文卷同现存的文卷结合起来, 并把所有叶节点排为单一序列; 而置换指丢弃现存文卷的数据并建立新的文卷内容。

A2.3.3 结构约束

根节点不应带有数据单元, 并不应有节点名; 根节点的所有子节点应是叶节点, 而且同时具有数据单元和节点名, 自根节点来的所有弧长都应是 1。

A2.3.4 动作约束

插入: 按照已定义的名字顺序, 带 FADU 标识 begin 的插入动作根据节点描述符(它必须被传送)中名字的位置在遍历序列中插入一个新的叶节点。如果已经存在具有相同名字值的 FADU 时, 新的 FADU 紧接着插在它们之后。插入之后的位置是 begin。

置换整个文卷: FADU 标识是 begin 且用整个有序平坦 FADU 置换根节点。

扩展或置换叶节点:该语义取决于被传送数据的形式。

a) 如果传送没有节点描述符的单个数据单元,那么由 FADU 标识寻址的叶节点内的数据单元被改变。

b) 如果传送了完整的叶 FADU 且 FADU 标识是带有单个成员的名字序列, current, previous, 或 next, 那么仅当在被传送的节点描述符里的节点名等于现存的节点名时,此动作才被执行。

c) 如果传送了完整的叶 FADU 且 FADU 标识是 begin, 那么在被传送的节点描述符里指定名字的那个节点被改变。

擦除:带 FADU 标识 begin 的擦除动作产生一个空文件;带有 FADU 标识 current, next, previous 或者只有单个成员的节点名序列的擦除动作删除所标识的叶节点。

注:如果存在具有相同节点名的多个节点时,则根据所使用的 FADU 标识或者将这些节点中按遍历次序的第一个或者将当前节点进行置换、扩展或者擦除。

A2.3.5 标识约束

与文卷动作相关联的 FADU 标识应是 begin, end, first, last, current, next, previous, 只有单个成员的节点名序列,或者大于或等于 1 的遍历序号中的一个。能与这些标识一同使用的动作在表 10 中给出。

注

- 1 如果在定位、读、插入、置换、或扩展动作之后或者在擦除动作之前,有一个或多个 FADU 与当前 FADU 同名时,则产生警告诊断,不同的诊断指明是否目前定位的 FADU 是在遍历序列中同此名的 FADU 中的最后一个。
- 2 FADU 标识 next 和 previous 主要是提供访问同名集中的 FADU。
- 3 不鼓励在此约束集中使用遍历序号。FADU 的次序由它们的节点名来确定,因此当文卷被修改之后,特定节点的遍历序号通常会被改变。

表 9 有序平坦约束集中的基本约束

约束集描述符	有序平坦
约束集标识符	{iso standard 8571 constraint-set(4) ordered-flat(3)}
节点名	所有名字应是相同类型,当引用该约束集时,应定义名字的类型和顺序
文卷访问动作	定位,读,插入,扩展,置换,擦除
限定的动作	没有
可用的访问上下文	HA,FA,UA
创建状态	不带有数据单元的根节点
打开后的位置	根节点
文卷的开头	根节点
文卷的结尾	无节点被选择,previous 给出在遍历序列中的最后节点,current 和 next 给出差错信息
读整个文卷	用 FADU 标识 begin 在访问上下文 FA 中读
写整个文卷(1):合并	传送一系列的叶 FADU,它们由用访问上下文 FA 读整个文卷而产生;用 FADU 标识 begin 和文卷访问动作插入来执行该传送。
写整个文件(2):替换	传送由一系列数据元素表示的 FADU,它们由用访问上下文 HA 读文卷而产生;用 FADU 标识 begin 和文卷访问动作置换来执行该传送

表 10 在有序平坦约束集中的标识约束

动作	Begin	End	First	Last	Current	Next	Previous	NodeSeq	结点数
定位	valid	valid	valid	valid	valid	valid	valid	valid	valid
读	whole		leaf	leaf	leaf	leaf	leaf	leaf	leaf
插入	leaf								
置换	whole				leaf	leaf	leaf	leaf	
扩展					leaf	leaf	leaf	leaf	
擦除	whole				leaf	leaf	leaf	leaf	

Node Seq=只有单个成员的节点名序列

A2.4 具有唯一名字的有序平坦约束集**A2.4.1 适用范围**

具有唯一名字的有序平坦约束集适用于由具有唯一名字的 FADU 序列构造的文卷,可根据各 FADU 的名字作出对它们的访问。

A2.4.2 基本约束

具有唯一名字的有序平坦约束集中的基本约束在表 11 中给出。

注:在写整个文卷项时,合并指将被传送的文卷同已存在的文卷结合起来,并把所有叶节点排为单一序列;而置换指丢弃现存的文卷数据并建立新的文卷内容。

A2.4.3 结构约束

根节点不应带有数据单元,并不应有节点名;根节点的所有子节点应是叶节点,而且应同时具有数据单元和节点名,从根节点来的所有弧长都应是 1。

A2.4.4 动作约束

插入:按照已定义的名字顺序,带 FADU 标识 begin 的插入动作根据节点描述符(它必须被传送)中名字的位置在遍历序列中插入一个新的叶节点。如果已经存在具有相同名字值的 FADU 时,则插入动作失败。插入之后的位置是 begin。

置换整个文卷:FADU 标识是 begin 且用整个具有唯一名字的有序平坦 FADU 置换根节点。

扩展或置换叶节点:该语义取决于被传送数据的形式。

a) 如果传送没有节点描述符的单个数据单元,那么由 FADU 标识寻址的叶节点内的数据单元被改变。

b) 如果传送了完整的叶 FADU 且 FADU 标识是只带单个成员的名字序列,current,previous,或 next,那么仅当在节点描述符里被传送的节点名等于现存的节点名时,此动作才被执行。

c) 如果传送了完整的叶 FADU 且 FADU 标识是 begin,那么在被传送的节点描述符里指定名字的那个节点被改变。

擦除:带 FADU 标识 begin 的擦除动作产生一个空文件;带有 FADU 标识为 current 或者只有单个成员的节点名序列的擦除动作删除所标识的叶节点。

A2.4.5 标识约束

与文卷动作相关联的 FADU 标识应是 begin,end,current,next,previous,只有单个成员的节点名序列,或者大于或等于 1 的遍历序号中的一个。能与这些标识一同使用的动作在表 12 中给出。

注:不鼓励在此约束集中使用遍历序号。FADU 的次序由它们的节点名来确定,因此当文卷被修改之后,特定节点的遍历序号通常会被改变。

表 11 具有唯一命名的有序平坦约束集中的基本约束

约束集描述符	具有唯一名字的有序平坦
约束集标识符	{iso standard 8571 constraint-set(4) ordered-flat-unique-names(4)}
节点名	所有名字应是相同类型,当引用该约束集时,应定义名字的类型和顺序
文卷访问动作	定位,读,插入,扩展,置换,擦除
限定的动作	没有
可用的访问上下文	HA,FA,UA
创建状态	不带有数据单元的根节点
打开后的位置	根节点
文卷的开头	根节点
文卷的结尾	无节点被选择,previous 给出在遍历序列中的最后节点,current 和 next 给出差错信息
读整个文卷	用 FADU 标识 begin 在访问上下文 FA 中读
写整个文卷(1):合并	传送一系列的叶 FADU,它们由用访问上下文 FA 读整个文卷而产生;用 FADU 标识 begin 和文卷访问动作插入来执行该传送。
写整个文件(2):替换	传送由一系列数据元素表示的 FADU,它们由用访问上下文 HA 读文卷而产生;用 FADU 标识 begin 和文卷访问动作置换来执行该传送

表 12 具有唯一命名的有序平坦约束集中的标识约束

动作	Begin	End	Current	Next	Previous	Node Seq	结点数
定位	valid	valid	valid	valid	valid	valid	valid
读	whole		leaf	leaf	leaf	leaf	leaf
插入	leaf						
置换	whole		leaf			leaf	
扩展			leaf			leaf	
擦除	whole		leaf			leaf	

Node Seq= 只有单个成员的节点名序列

A2.5 有序分级约束集

A2.5.1 适用范围

有序分级约束集适用于分级构造的文卷,并且其中每个节点的所有子节点都在遍历序列中据已定义的顺序按它们名字的位置排序。

A2.5.2 基本约束

有序分级约束集中的基本约束在表 13 中给出。

A2.5.3 结构约束

没有结构约束,分级文卷模型的全部通用性都是可用的。

A2.5.4 动作约束

插入:新节点插入到已定位节点的子节点中,并在定位时,使该节点的所有子节点名仍有序;被插入的节点名要与该已定位节点的所有现存子节点名都不同。

其他的动作:其他的动作完全按第 11 章中的规定。

A2.5.5 标识约束

与文卷动作相关联的 FADU 标识应是 begin, end, current, next, previous, NodeName, 或节点名序列。能与这些标识一同使用的动作在表 14 中给出。

注

- 1 当对整个文卷实施动作时,一个节点名的空序列用来对根节点编址。
- 2 next 和 previous 一般被用在访问上下文 FS 中,或用于访问一个父节点的一系列的叶结点。

表 13 有序分级约束集中的基本约束

约束集描述符	有序分级
约束集标识符	{iso standard 8571 constraint-set(4) ordered-hierarchical(5)}
节点名	所有名字应是相同类型;每一个 FADU 的名字相对于它的姐妹节点名是唯一的;当引用这个约束集时,应定义名字的类型和顺序
文卷访问动作	定位,读,插入,置换,扩展,擦除
限定的动作	没有
可用的访问上下文	HA,HN,FA,FL,FS,UA,US
创建状态	不带数据单元的根节点
打开后的定位	文卷的开头
文卷的开头	无节点被选择,next 给出按遍历序列的第一个节点(根节点);current 和 previous 给出差错信息
文卷的结尾	无节点被选择,previous 给出按遍历序列的最后一个节点,current 和 next 给出差错信息
读整个文卷	在带有由一个节点名空序列构成的 FADU 标识的访问上下文 HA 中读
写整个文卷	传送的 FADU 为由在访问上下文 HA 中读文卷所产生的数据元素的序列。用由一个节点名空序列构成的 FADU 标识和文卷访问动作置换来完成传送。

表 14 有序分级约束集中的标识约束

动作	Begin	End	Current	Next	Previous	NodeName	NodeSeq
定位	valid	valid	valid	valid	valid	valid	valid
读			valid	valid	valid	valid	valid(注 1)
插入			valid			valid	valid
置换			valid			valid	valid(注 1)
扩展			valid			valid	valid
擦除			valid			valid	valid(注 1)

NodeSeq=节点名的一个序列

A2.6 通用分级约束集

A2.6.1 适用范围

通用分级约束集适用于分级结构的文卷,并具有通用结构的所有通用性。

A2.6.2 基本约束

通用分级约束集中的基本约束在表 15 中给出。

A2.6.3 结构约束

没有结构约束,分级文卷模型的所有通用性都是可用的。

A2.6.4 动作约束

插入:有两种形式的插入动作。

常规动作——姐妹节点插入:创建的 FADU 按前序遍历序列放在当前位置之后,并使它链到它父节点的弧具有在节点描述符中所规定的长度。插入的 FADU 和当前位置具有同一个父节点。

如果当前位置是文卷的根,则不能进行姐妹节点插入。

该动作后的当前位置是新插入的 FADU 的根节点。

限定动作——子节点插入:当开始一系列限定动作时,记录下当前位置并建立一个新位置,这个新位置在遍历序列中位于 Previous 和下一个节点之间,即位于先前定位的节点的第 1 个子节点(如果有的话)之前。

然后用与常规插入动作姐妹节点插入相同的语义来执行这些限定动作,但使用这个临时的位置。

在这些限定动作结束时,根据在这一系列限定动作开始前所记录的位置重新恢复当前位置。

其他的动作:其他的动作完全按第 11 章中的规定。

注:如果多个节点具有相同的节点名,则根据使用的 FADU 标识或者将按遍历序列中的第一个节点,或者将当前节点进行置换、扩展或删除。

A2.6.5 标识约束

与文卷动作相关联的 FADU 标识应是 begin, end, current, next, previous, NodeName, 或节名序列。能与这些标识一同使用的动作在表 16 中给出。

表 15 通用分级约束集中的基本约束

约束集描述符	通用分级
约束集标识符	{iso standard 8571 constraint-set(4) general-hierarchical(6)}
节点名	当引用这个约束集时,应定义这个名字的类型;在分级的同一级上的各个名字应是相同类型的
文卷访问动作	定位,读,插入,置换,扩展,擦除
限定动作	插入:常规动作——姐妹节点插入 限定动作——子节点插入
可用访问上下文	HA,HN,FA,FL,FS,UA,US
创建状态	不带数据单元的根节点
打开后的定位	文卷的开头
文卷的开头	无节点被选择;next 给出按遍历序列的第一个节点(根节点);current 和 previous 给出差错信息
文卷的结尾	无节点被选择;previous 给出按遍历序列的最后一个节点;current 和 next 给出差错信息
读整个文卷	在带有由一个节点名空序列构成的 FADU 标识的访问上下文 HA 中读
写整个文卷	传送的 FADU 为由在访问上下文 HA 中读文卷所产生的数据元素的序列。用由一个节点名空序列构成的 FADU 标识和文卷访问动作置换来完成传送

表 16 通用分级约束集中的标识约束

动作	Begin	End	Current	Next	Previous	NodeName	NodeSeq
定位	valid	valid	valid	valid	valid	valid	valid
读			valid	valid	valid	valid	valid(注 1)
插入			valid	valid	valid	valid	valid
置换			valid	valid	valid	valid	valid(注 1)
扩展			valid	valid	valid	valid	valid
擦除			valid	valid	valid	valid	valid(注 1)

NodeSeq = 一个节点名序列

注

- 1 当对整个文卷实施动作时,用一个节点名的空序列来对根节点编址。
- 2 如果在定位、读、插入、置换、或扩展动作之后或者擦除动作之前,有一个或多个 FADU 与当前 FADU 同名,则产生警告诊断,不同的诊断指明是否目前定位的 FADU 是在遍历序列中同此名的 FADU 中的最后一个。
- 3 FADU 标识 next 和 previous 主要是提供访问同名集中的 FADU。
- 4 next 和 previous 一般被用在访问上下文 FS 中,或用于访问一个父节点的一系列的叶节点。

A2.7 具有唯一名字的通用分级约束集

A2.7.1 适用范围

具有唯一名字的通用分级约束集适用于分级结构文卷,除了要求任一给定父节点的子节点名必须唯一外,具有通用结构的所有通用性。

A2.7.2 基本约束

具有唯一名字的通用分级约束集中的基本约束列于表 17 中。

A2.7.3 结构约束

没有结构约束,分级文卷模型的所有通用性都是可用的。

A2.7.4 动作约束

插入:有两种形式的插入动作。被插入的节点名必须与它现存的姐妹节点名不同。

常规动作——姐妹节点插入:创建的 FADU 按前序遍历序列放在当前位置之后,并使它链到它父节点的弧具有在节点描述符中所规定的长度。插入的 FADU 和当前位置具有同一个父节点。

如果当前位置是文卷的根,则不能执行姐妹插入。

该动作后的当前位置是新插入的 FADU 的根节点。

限定动作——子节点插入:当开始一系列限定动作时,记录下当前位置并建立一个新位置,这个新位置在遍历序列中位于 Previous 和下一个节点之间,即位于先前定位的节点的第 1 个子节点(如果有的话)之前。

然后用与常规插入动作姐妹节点插入相同的语义来执行这些限定动作,但使用这个临时的位置。

在这些限定动作结束时,根据在这一系列限定动作开始时所记录的位置重新恢复当前位置。

其它的动作:其它的动作完全按第 11 章中的规定。

A2.7.5 标识约束

与文卷动作相关联的 FADU 标识应是 begin, end, current, next, previous, NodeName, 或节点名序列。能与这些标识一同使用的动作在表 18 中给出。

注

- 1 当对整个文卷实施动作时,用一个节点名的空序列来对根节点编址。
- 2 next 和 previous 一般被用在访问上下文 FS 中,或用于访问一个父节点的一系列的叶节点。

表 17 具有唯一命名的通用分级约束集中的基本约束

约束集描述符	具有唯一名字的通用分级
约束集标识符	{iso standard 8571 constraint-set(4) general-hierarchical-unique-name(7)}
节点名	当引用这个约束集时,应定义这个名字的类型;在分级的同一级上的各个名字应是相同类型的。任一特定父节点的子节点名应是相同类型的
文卷访问动作	定位,读,插入,置换,扩展,擦除
限定动作	插入:常规动作——姐妹节点插入 限定动作——子节点插入
可用访问上下文	HA,HN,FA,FL,FS,UA,US
创建状态	不带数据单元的根节点
打开后的定位	文卷的开头
文卷的开头	无节点被选择;next 给出按遍历序列的第一个节点(根节点);current 和 previous 给出差错信息
文卷的结尾	无节点被选择;previous 给出按遍历序列的最后一个节点;current 和 next 给出差错信息
读整个文卷	在带有由一个节点名空序列构成的 FADU 标识的访问上下文 HA 中读
写整个文卷	传送的 FADU 为由在访问上下文 HA 中读文卷所产生的数据元素的序列。用由一个节点名空序列构成的 FADU 标识和文卷访问动作置换来完成传送

表 18 具有唯一命名的通用分级约束集中的标识约束

动作	Begin	End	Current	Next	Previous	NodeName	NodeSeq
定位	valid	valid	valid	valid	valid	valid	valid
读			valid			valid	valid(注 1)
插入			valid			valid	valid
置换			valid			valid	valid(注 1)
扩展			valid			valid	valid
擦除			valid			valid	valid(注 1)

NodeSeq=一个节点名序列

附 录 B

(标准的附录)

文 件 类 型

本附录包含了一些公共文件类型定义以允许 FTAM 实现之间的交互工作。

注:在 GB/T 16505 的这部分包含这些定义以留待 ISO 文件类型登记机构的建立。一旦建立起这个机构,预计这些定义将被包含在文件类型登记中,从而 GB/T 16505 这部分的未来版本将可略去这个附录。FTAM 的使用不一定要求使用文件类型;文卷的内容可由一个抽象语法和适用于该文卷模型的一个约束集来规定。

B1 无结构正文文卷文件类型**1 项号:FTAM-1****2 信息客体**

表 19 无结构正文文件类型中的信息客体

文件类型名	{iso standard 8571 document-type(5) unstructured-text(1)} “ISO FTAM unstructured text”
抽象语法名 a. asname1 的名	{iso standard 8571 abstract-syntax(2) unstructured-text(3)} “FTAM unstructured text abstract syntax”
传送语法名	{joint-iso-ccitt asnl(1) basic-encoding(1)} “Basic encoding of a single ASN.1 type”
参数语法	PARAMETERS ::= SEQUENCE { universal-class-number [0] IMPLICIT INTEGER OPTIONAL, maximum-string-length [1] IMPLICIT INTEGER OPTIONAL, string-significance [2] IMPLICIT INTEGER OPTIONAL { variable (0), fixed (1), not-significant (2) } OPTIONAL }
文卷模型	{iso standard 8571 file-model(3) hierarchical(1)} “FTAM hierarchical file model”
约束集	{iso standard 8571 constraint-set(4) unstructured(1)} “FTAM unstructured constraint set”
文卷内容	Datatype1 ::= CHOICE { PrintableString --通用类 19--, TeletexString --通用类 20--, VideotexString --通用类 21--, IA5String --通用类 22--, GraphicString --通用类 25--, VisibleString --通用类 26--, GeneralString --通用类 27-- }

3 范围

这些文件类型为存储、为 FTAM 传送和访问以及为用 JTM 传送定义了一个文卷的内容。

4 引用标准

GB/T 5261—94 信息处理 七位和八位编码字符集用的控制功能(idt ISO 6429:1988)

GB/T 16505—1996 信息处理系统 开放系统互连 文卷传送、访问和管理(idt ISO 8571:1988)

ISO 8832:1989 信息处理系统 开放系统互连 作业传送与操纵的基本类协议规范

5 定义

本定义使用在 GB/T 16505.1 中定义的数据元素、数据单元和文卷访问数据单元等术语。

5.1 字符串:character string

来自某些规定的字符集的一个有序的 0 个、1 个或多个字符的序列。

5.2 图形字符:graphics character

任何登记过的字符集中的字符,如在 ISO 国际字符集登记中 G0、G1、G2 或 G3 集的字符,或空格字符。

5.3 格式控制符:format effector

控制在打印或显示设备上信息格式和位置的一种控制功能。公共的格式控制字符是 BS、CR、FF、HT、LF 和 VT。

6 缩略语

FTAM 文卷传送、访问和管理

JTM 作业传送和操纵

7 文件的语义

这种文件包含一个仅含有 0 个、1 个或多个字符串的文卷访问数据单元。这些元素的次序是有意义的,这些字符串的语义不由这个文件类型来定义。

这种文件结构采取如同无结构约束集(见表 19)约束的 FTAM 分级文卷模型所允许的形式,这些定义在 GB/T 16505.2 中出现。

每个字符串由 ASN.1(GB/T 16262)字符集类型定义的字符集中的字符组成,该类型的通用类号由 Universal-class-number 参数给定。如果 Universal-class-number 不存在,则缺省值为 GraphicString (25);即该字符串可包含来自任何登记(在带转义序列的编码字符集的国际登记中)用作为 G0、G1、G2、或 G3 字符集(加上空格字符)中的字符。

除在这里说明的以外,这个定义没有隐含其他大小或长度限制。每个字符串具有由 maximum-string-length 参数给出的字符数所确定的一个长度,如果 maximum-string-length 参数不存在,则缺省规则是字符串无界。

注:长度限制引用的是来自适用字符集中的字符数,而不是编码中的八位位组数,也不是行长,对于该文件的任何应用它们是不同的。

字符串准确的有效位是由 String-significance 参数决定的。如果它的值是“variable”,或这个参数不存在,则这个字符串的长度小于或等于给定的长度。如果这个值是“fixed”,则每个字符串的长度严格地等于给定的长度。如果这个值是“not-significant”,则当文卷被存储时字符串的边界不需要保留而且对该文件的语义不起作用。

如果在一个字符图象设备(超出 GB/T 16505 的范围)上解释该文件,则该解释取决于使用中的字符集:

a) 如果字符集包括格式控制符,则它们应按 GB/T 5261 中的规定解释;串结束和文卷访问数据单元的结束没有格式上的意义。

b) 如果字符集不包括格式控制符,则在任何使用中每个字符串的结束解释为隐含回车和换行格式动作。文卷访问数据单元的结束没有任何格式意义,它只表示其最后一个字符串的结束。

8 抽象语法结构

文件的抽象语法结构是一系列字符串,其中每种类型均具有用 universal-class-number 参数指出的 ASN.1 字符串类型。

9 传送定义

9.1 数据类型定义

文卷包含在表 19 中定义的 0 个或多个 Datatype1 类型的值,在这个数据类型的选择是由 univer-

sal-class-number 参数,或它的缺省值来决定的。

9.2 表示数据值

该文件作为一系列表示数据值来传送,每个表示数据值应包含 ASN.1 数据类型 Datatype1 的一个值,它携带从文件来的一个字符串。每一个字符应用由 universal-class-number 参数标识的一种字符集来传输。

用(任意)同一个已建立的表示上下文传输所有值,该表示上下文支持在表 19 中说明的抽象语法名“asname1”。

注:特定的载体标准可能对要使用的表示上下文有附加约束,此时上述原则允许选择表示上下文。

P-DATA 原语之间的边界是在传输时由发送实体在本地选择,并且不携带文件类型的任何语义;支持这个文件类型的接收者应可接受具有任何允许的传送选项的该类文件。

9.3 表示数据值序列

表示数据值序列与文卷数据单元中的字符串序列相同。

10 传送语法

支持这些文件类型的实现对所传送的所有表示数据值应支持在表 19 中命名的传送语法产生规则。

实现还可有选择地支持其它传送语法。

11 ASE 特定规范

11.1 GB/T 16505(FTAM)

11.1.1 简化和松弛

11.1.1.1 字符集松弛

这个操作会丢失在文件类型标识中的显式信息。

如果结果文件类型允许初始的文件类型的所有字符的话,一个 FTAM-1 类型的文件可松弛为一个具有不同的 universal-class-number 参数值或没有参数值的 FTAM-1 类型的文件。如果这个松弛包含一个格式控制符集,并且在简化以前这些格式控制符都不存在,则字符“回车”和“换行”应加到每个字符串的末端。

注:如果字符“回车”和“换行”不是格式控制符集中的一部分,则格式化动作可能由“新行”表示;如果没有定义“新行”的表示,则可用实现的某些其他特定选择来表示。

11.1.1.2 字符串长度松弛

这个操作丢失在文件类型标识中的显式信息。

一个 FTAM-1 类型的文件可松弛为另一个具有较大 maximum-string-length 参数值,或没有 maximum-string-length 参数(无界字符串)的 FTAM-1 类型的文件。

11.1.2 扩展操作

当扩展操作被用于 FTAM-1 文件类型的数据单元时,被传送的数据应是具有一个 FTAM-1 文件类型,其参数等于初始 FTAM-1 文件类型中的这些参数。结果文件由原 FTAM-1 文件类型的字符串和紧跟着的新 FTAM-1 文件类型的字符串组成。新老字符串之间的边界在新的文件中是不可见的。

11.1.3 置换操作

当置换操作被用于一个 FTAM-1 文件的根 FADU 时,被传送的东西应是具有相同参数值的某个 FTAM-1 文件。

11.2 ISO 8832(JTM)

下列几条适用于宣称支持 FTAM-1 的任何 JTM 实现。

11.2.1 支持其他文件类型

文件类型 JTM-1 应也被支持。

注:这些文件类型在语义上是相同的。

11.2.2 长度和参数

在基于来自文件中规定的字符集中字符总数的可能实现限制之下,实现应能够处理具有各种参数值的任何 FTAM-1 文件类型。

注:关于“支持”的规范见 ISO 8832,特别是与字符集的关系。

实现的限制应允许使用至多包含 64 000 个字符的文件。这些限制应在协议实现的一致性声明中描述。

11.2.3 拼接

应支持 FTAM-1 类型文件本身的拼接,并且应产生一个相同类型的包含组合的字符串序列的文件。

注:原文件的边界在结果文件中仅看作是两个字符串之间的常规边界。

11.2.4 求精

11.2.4.1 字符集和长度求精

由 J-GIVE 使用 FTAM-1 文件类型请求的任何文件应可通过处理从存储的 FTAM-1 文件类型中获得,这个 FTAM-1 文件是按 11.1.1.1 或 11.1.1.2 或者同时按这两条中的规则所获得的一个松弛。这个处理可能产生一个文件,也可能产生一个诊断。详见 ISO 8832。

11.2.5 松弛

对于 J-GIVE 使用 FTAM-1 文件类型的任何文件,若该文件类型是被存储的文件类型的一个松弛,则应按所请求的 FTAM-1 文件类型提供。

注:在 J-GIVE 响应中的值从不紧缩。

B2 顺序正文文卷文件类型

1 项号:FTAM-2

2 信息客体

表 20 顺序正文文卷文件类型中的信息客体

文件类型名	{iso standard 8571 document-type(5) sequential-text(2)} “ISO FTAM sequential text”
抽象语法名 a. asname1 的名	{iso standard 8571 abstract-syntax(2) unstructured-text(3)} “FTAM unstructured text abstract syntax”
b. asname2 的名	{iso standard 8571 abstract-syntax(2) ftam-fadu(2)} “FTAM FADU”
传送语法名	{joint-iso-ccitt asnl(1) basic-encoding(1)} “Basic encoding of a single ASN.1 type”
参数语法	PARAMETERS::=SEQUENCE{universal-class-number[0] IMPLICIT INTEGER OPTIONAL, maximum-string-length[1] IMPLICIT INTEGER OPTIONAL, string-significance[2] IMPLICIT INTEGER {variable(0), fixed(1), not-significant(2)} OPTIONAL}
文件模型	{iso standard 8571 file-model(3) hierarchical(1)} “FTAM hierarchical file model”
约束集	{iso standard 8571 constraint-set(4) sequential-flat(2)} “FTAM sequential flat constraint set”

表 20(完)

文件类型名	{iso standard 8571 document-type(5) sequential-text(2)} “ISO FTAM sequential text”
文件内容	
Datatype1::=CHOICE {PrintableString --通用类 19--,	
TeletexString --通用类 20--,	
VideotexString --通用类 21--,	
IA5String --通用类 22--,	
GraphicString --通用类 25--,	
VisibleString --通用类 26--,	
GeneralString --通用类 27--}	
Datatype2::=	Node-Descriptor-Data-Element

3 范围

这些文件类型为存储、为 FTAM 传送和访问以及为用 JTM 传送定义了一个文卷的内容。

4 引用标准

GB/T 5261—94 信息处理 七位和八位编码字符集用的控制功能(idt ISO 6429:1988)

GB/T 16505—1996 信息处理系统 开放系统互连 文卷传送、访问和管理(idt ISO 8571)

ISO 8832:1989 信息处理系统 开放系统互连 作业传送与操纵的基本类协议规范

5 定义

本定义使用在 GB/T 16505.1 中定义的数据元素、数据单元和文卷访问数据单元等术语。

5.1 字符串:character string

来自某些规定的字符集的一个有序的 0 个、1 个或多个字符的序列。

5.2 图形字符:graphics character

任何登记过的字符集中的字符,如在 ISO 国际字符集登记中的 G0、G1、G2 或 G3 集的字符,或空格字符。

5.3 格式控制符:format effector

控制在打印或显示设备上信息格式和位置的一种控制功能。公共的格式控制字符是 BS、CR、FF、HT、LF 和 VT。

6 缩略语

FTAM 文卷传送、访问和管理

JTM 作业传送和操纵

7 文件的语义

这种文件包含 0 个、1 个或多个文卷访问数据单元。每个文卷访问数据单元包含 0 个、1 个或多个字符串。各个元素的次序是有意义的。这些字符串的语义不规定于这个文件类型。

这种文件结构取由顺序平坦约束集(见表 20)约束的 FTAM 分级文卷模型所允许的形式,这些定义出现在 GB/T 16505.2 中。

每个字符串由 ASN.1(GB/T 16262)字符集类型定义的字符集中的字符组成,该类型的通用类号由“Universal-class-number”参数给定,如果“Universal-class-number”没有出现,则缺省为 GraphicString(25);即该字符串可包含来自任何登记(在带转义序列的编码字符集的国际登记中)用作为 G0、G1、G2、或 G3 字符集(加上空格字符)中的字符。

除在这里说明的以外,这个定义没有隐含其他大小或长度限制。每个字符串具有由 maximum-string-length 参数给出的字符数所确定的一个长度,如果 maximum-string-length 参数不存在,则缺省

规则是字符串无界。

注：长度限制针对的是来自适用字符集中的字符数，而不是编码中的八位位组数，也不是行长，对于该文件的任何应用它们是不同的。

字符串准确的有效位是由 String-significance 参数决定的。如果它的值是“variable”，或这个参数不存在，则这个字符串的长度小于或等于给定的长度。如果这个值是“fixed”，则每个字符串的长度严格地等于给定的长度。如果这个值是“not-significant”，则当文卷被存储时字符串的边界不需要保留且对该文件的语义不起作用。

如果在一个字符图像设备(超出 GB/T 16505 的范围)上解释该文件，则该解释取决于使用中的字符集：

a) 如果字符集包括格式控制符，则它们应按 GB/T 5261 中的规定解释；串结束和文卷访问数据单元的结束没有格式上的意义。

b) 如果字符集不包括格式控制符，则在任何使用中每个字符串的结束解释为隐含回车和换行格式动作。文卷访问数据单元的结束没有任何格式意义，它只表示其最后一个字符串的结束。

8 抽象语法结构

文件的抽象语法结构是由在 GB/T 16505 中 ASN.1 模块 ISO 8571-FADU 和 ISO 8571-CONTENTS 中所定义的分级结构文卷，其中每一个文卷内容数据元素具有在 GB/T 16505 中文件类型登记项 FTAM-1 中定义的“ISO FTAM unstructured text”文件的抽象语法结构。

9 传送定义

9.1 数据类型定义

文卷由具有下列任一类型的数据值组成

a) 在表 20 中定义的 Datatype1，其中数据类型的选择由 universal-class-number 参数或它的缺省值所决定。

b) 表 20 中定义的 Datatype2，它们是在 ASN.1 模块 ISO 8571-FADU 中被说明为 Data-Element 的 ASN.1 数据类型。

9.2 表示数据值

该文件作为一系列表示数据值传送，其中每个值是下列两种情况之一：

a) ASN.1 数据类型 Datatype1 的值，它携带从文件来的一个字符串。每一个字符应用由 universal-class-number 参数标识的一种字符集来传输。用(任意)同一个已建立的表示上下文传输所有值，该表示上下文支持在表 20 中说明的抽象语法名“asname1”。

b) ASN.1 数据类型 Datatype2 的值。用(任意)同一个已建立的表示上下文传输所有值，该表示上下文支持在表 20 中说明的抽象语法名“asname2”。

注

1 特定的载体标准可能对要使用的表示上下文有附加约束，此时上述原则允许选择表示上下文。

2 在本项中定义的任何文件类型或者不使用 Datatype2，或者以 Datatype2 开始传输。

P-DATA 原语之间的边界是在传输时由发送实体在本地选择的，并且不携带文件类型的任何语义；支持这个文件类型的接收者将接受一个具有任何允许的传送选项的该类文件。

9.3 表示数据值序列

当根据 GB/T 16505.2 中分级文卷模型定义将文卷平坦化时，类型(a)的表示数据值序列和类型(a)与(b)的表示数据值序列分别与一个数据单元中的，以及分级结构中诸数据单元的字符串序列相同。

10 传送语法

支持这些文件类型的实现对所传送的所有表示数据值应支持在表 20 中命名的传送语法产生规则。实现还可有选择地支持其他传送语法。

11 ASE 特定规范

11.1 GB/T 16505(FTAM)

11.1.1 简化和松弛

11.1.1.1 结构简化

此种简化丢失信息。

文件类型 FTAM-2 可被简化成文件类型 FTAM-1。结果文件含有的数据值序列与按访问上下文 UA 访问已有结构的文本文卷所得的结果相同。这就是说,只有在抽象语法“asname1”中的表示数据值才存在。

注:文卷访问数据单元之间的边界保持为字符串之间的边界,但是任何赋与它的特殊意义都将失去。

11.1.1.2 字符集松弛

这个操作丢失在文件类型标识中的显式信息。

如果结果文件类型允许初始文件类型的所有字符的话,一个 FTAM-2 类型的文件可松弛为一个具有不同的 universal-class-number 参数值或没有参数值的 FTAM-2 类型的文件。如果这个松弛包含加入一个格式控制符集,并且在简化以前这些格式控制符都不存在,则字符“回车”和“换行”应被加到每个字符串的末端。

注:如果字符“回车”和“换行”不是格式控制符集中的一部分,则格式化动作可能由“新行”表示;如果没有定义“新行”的表示则可用实现的某些其他特定选择来表示。

11.1.1.3 字符串长度松弛

这个操作丢失在文件类型标识中的显式信息。

一个 FTAM-2 类型的文件可松弛为另一个具有较大 maximum-string-length 参数值,或没有 maximum-string-length 参数(无界字符串)的 FTAM-2 类型的文件。

11.1.2 访问上下文选择

FTAM-2 类型的文件可在顺序平坦约束集中定义的任一访问上下文中访问。在每种情形下传送的表示数据单元就是那些从在 GB/T 16505.2 中为该访问上下文定义的结构元素中得到的数据单元。

在访问上下文 FA 中,结果文件遵从 FTAM-2 类型。在访问上下文 UA 中,结果文件遵从 FTAM-1 类型。

11.1.3 插入操作

当插入操作应用在文卷尾部时,传送的东西应是一系列 FADU,这些 FADU 应以在访问上下文 FA 中用相同参数读任一 FTAM-2 类型文件的方式生成。

11.2 ISO 8832(JTM)

下列几条适用于宣称支持 FTAM-2 的任何 JTM 实现。

11.2.1 支持其他文件类型

文件类型 FTAM-1 和 JTM-1 应也被支持。

注:这些文件类型在语义上是相同的。

11.2.2 长度和参数

基于来自文件中规定的字符集中字符总数的可能实现限制,实现应能够处理具有各种参数值的任何 FTAM-2 文件类型。

注:关于“支持”的规范见 ISO 8832,特别是与字符集的关系。

实现的限制应允许使用至多包含 64 000 个字符的文件,这些限制应在协议实现的一致性声明中描述。

11.2.3 拼接

应支持 FTAM-2 类型文件本身的拼接,并且应产生一个包含组合的字符串序列的相同类型的文件。

注:原文件的边界在结果文件中仅看作是两个字符串之间的常规边界。

11.2.4 求精

11.2.4.1 字符集和长度求精

由 J-GIVE 使用 FTAM-2 文件类型请求的任何文件应通过处理从存储的 FTAM-2 文件类型中获得,这个 FTAM-2 文件是按 11.1.1.2 或 11.1.1.3 或者同时按这两条中的规则所获得的一个松弛。这个处理可能产生一个文件,或产生一个诊断。详见 ISO 8832。

11.2.4.2 结构求精

任一由 J-GIVE 用 FTAM-2 文件类型请求的文件,应从任一存储的 FTAM-1 文件类型中通过把整个 FTAM-1 文件作为 FTAM-2 文件的单个数据单元(唯一的数据单元)而得到。

11.2.5 结构简化

任何由 J-GIVE 用 FTAM-1 文件类型请求的文件应通过应用在 11.1.1.1 中规定的结构简化从任一 FTAM-2 文件类型中得到。

11.2.6 松弛

对于 J-GIVE 使用 FTAM-2 文件类型请求的任何文件,若该文件类型是被存储的文件类型的一个松弛,则应按所请求的 FTAM-2 文件类型提供。

注:在 J-GIVE 响应中的值从不紧缩。

B3 无结构二进制文卷文件类型

1 项号:FTAM-3

2 信息客体

表 21 无结构二进制文件类型中的信息客体

文件类型名	{iso standard 8571 document-type(5) unstructured binary(3)} “ISO FTAM unstructured binary”
抽象语法名 a. asname1 的名	{iso standard 8571 abstract-syntax(2) unstructured-binary(4)} “FTAM unstructured binary abstract syntax”
传送语法名	{joint-iso-ccitt asnl(1) basic-encoding(1)} “Basic encoding of a single ASN.1 type”
参数语法	PARAMETERS::=SEQUENCE{maximum-string-length[1] IMPLICIT INTEGER OPTIONAL, string-significance[2] IMPLICIT INTEGER {Variable(0),fixed(1),not-significant(2)} OPTIONAL}
文卷模型	{iso-standard 8571 file-model(3) hierarchical(1)} “FTAM hierarchical file model”
约束集	{iso-standard 8571 constraint-set(4) unstructured(1)} “FTAM unstructured constraint set”
文卷内容	Datatype1::=OCTET STRING

3 主题内容与适用范围

这些文件类型为存储、为 FTAM 传送和访问以及为用 JTM 传送定义了一个文卷的内容。

4 引用标准

GB/T 16505—1996 信息处理系统 开放系统互连 文卷传送、访问和管理(idt ISO 8571:1988)
ISO 8832 信息处理系统 开放系统互连 作业传送与操纵的基本类协议规范

5 定义

本定义使用在 GB/T 16505.1 中定义的数据元素,数据单元和文卷访问数据单元等术语。

5.1 二进制串;binary string

一个由零个、一个或多个八位位组组成的有序序列。

6 缩略语

FTAM 文卷传送、访问和管理

JTM 作业传送和操纵

7 文件的语义

这种文件包含一个仅含有 0 个、1 个或多个二进制串的文卷访问数据单元。这些元素的次序是有意
义的,这些二进制串的语义不由这个文件类型来定义。

这种文件结构采取如同无结构约束集(见表 21)约束的 FTAM 分级文卷模型所允许的形式,这些
定义出现在 GB/T 16505.2 中。

每一二进制串由值为 0 到 255 的八位位组组成,赋与这些值的意义不受该文件类型限制。

除在这里说明的之外,这个定义没有隐含其他大小或长度限制。每个二进制串具有由 maximum-
string-length 参数给出的字符数所确定的一个长度,如果 maximum-string-length 参数不存在,则缺省
规则是二进制串无界。

二进制串准确的有效位是由 String-significance 参数决定的。如果它的值是“variable”,或这个参数
不存在,则这个二进制串的长度小于或等于给定的长度。如果这个值是“fixed”,则每个二进制串的长度
严格地等于给定的长度。如果这个值是“not-significant”,则当文卷被存储时二进制串的边界不需要保
留且对该文件的语义不起作用。

8 抽象语法结构

本文件的抽象语法结构是一个具有类型 OCTET STRING 的 ASN.1 数据类型序列。

9 传送定义

9.1 数据类型定义

文卷中包含定义在表 21 中的 0 个或多个 Datatype1 类型的值。

9.2 表示数据值

该文件作为一系列表示数据值来传送,每个表示数据值应包含 ASN.1 数据类型 Datatype1 的一个
值,它携带文件中的一个二进制串。

用(任意)同一个已建立的表示上下文中传输所有值,该表示上下文支持在表 21 中说明的抽象语法
名“asname1”。

注:特定的载体标准可能对要使用的表示上下文有附加约束,此时上述原则允许选择表示上下文。

P-DATA 原语之间的边界是在传输时由发送实体在本地选择的,并且不携带文件类型的任何语
义;支持这个文件类型的接收者将接受一个具有任何允许的传送选项的该类文件。

9.3 表示数据值序列

表示数据值序列与文卷中数据单元中的二进制串序列相同。

10 传送语法

支持这些文件类型的一种实现对传送的所有表示数据值应支持在表 21 中命名的传送语法产生规
则。

一个实现还可有选择地支持其他传送语法。

11 ASE 特定规范

11.1 GB/T 16505(FTAM)

11.1.1 简化和松弛

11.1.1.1 串长度松弛

这个操作丢失在文件类型标识中的显式信息。

一个 FTAM-3 类型的文件可松弛为另一个具有较大 maximum-string-length 参数值,或没有 maximum-string-length 参数(无界字符串)的 FTAM-3 类型的文件。

11.1.2 扩展操作

当扩展操作被用于 FTAM-3 文件类型的数据单元时,被传送的数据应具有 FTAM-3 文件类型,其参数等于初始 FTAM-3 文件类型中的那些参数。结果文件由原 FTAM-3 文件类型的二进制串紧跟着新 FTAM-3 文件类型的二进制串组成。新老二进制串之间的边界在新的文件中是不可见的。

11.1.3 置换操作

当置换操作被用于一个 FTAM-3 文件的根 FADU 时,被传送的东西应是具有相同参数值的某个 FTAM-3 文件。

11.2 ISO 8832(JTM)

下列几条适用于宣称支持 FTAM-3 的任何 JTM 实现。

11.2.1 长度和参数

基于八位位组总数的可能实现限制,实现应能够处理具有各种参数值的任何 FTAM-3 文件类型。

注:关于“支持”的规范见 ISO 8832。

实现的限制应允许使用至多包含 64 000 个八位位组的文件。这些限制应在协议实现的一致性声明中描述。

11.2.2 拼接

应支持 FTAM-3 类型文件本身的拼接,并且应产生一个包含组合的二进制串序列的相同类型的文件。

注:原文件的边界在结果文件中仅看作是两个二进制串之间的常规边界。

11.2.3 求精

11.2.3.1 字符集和长度求精

由 J-GIVE 通过使用 FTAM-3 文件类型请求的任何文件应从存储的 FTAM-3 文件类型中获得,这个 FTAM-3 文件是按 11.1.1.1 中的规则处理所获得的一个松弛。这个处理可能产生一个文件,或产生一个诊断。详见 ISO 8832。

11.2.4 松弛

对于 J-GIVE 通过使用 FTAM-3 文件类型请求的任何文件,若该文件类型是被存储的文件类型的一个松弛,则应按所请求的 FTAM-3 文件类型提供。

注:在 J-GIVE 响应中的值从不紧缩。

B4 顺序二进制文卷文件类型

1 项号:FTAM-4

2 信息客体

表 22 顺序二进制文件类型中的信息客体

文件类型	{iso standard 8571 document-type(5) sequential-binary(4)} “ISO FTAM sequential binary”
抽象语法名 a. asname1 的名	{iso standard 8571 abstract-syntax(2) unstructured-binary(4)} “FTAM unstructured binary abstract syntax”

表 22(完)

文件类型	{iso standard 8571 document-type(5) sequential-binary(4)} “ISO FTAM sequential binary”
b. asname2 的名	{iso standard 8571 abstract-syntax(2) fadu(2)} “FTAM FADU”
传送语法名	{joint-iso-ccitt asnl(1) basic-encoding(1)} “Basic encoding of a single ASN.1 type”
参数名 PARAMETERS ::= SEQUENCE { maximum-string-length[1] IMPLICIT INTEGER OPTIONAL, string-significance[2] IMPLICIT INTEGER {variable(0), fixed(1), not-significant(2)} OPTIONAL }	
文件模型	{iso standard 8571 file-model(3) hierarchical(1)} “FTAM hierarchical file model”
约束集	{iso standard 8571 constraint-set(4) sequential-flat(2)} “FTAM sequential flat constraint set”
文件内容 Datatype1 ::= OCTET STRING Datatype2 ::= Node-Descriptor-Data-Element	

3 范围

这些文件类型为存储、为 FTAM 传送和访问以及为用 JTM 传送定义了一个文卷的内容。

4 引用标准

GB/T 16505—1996 信息处理系统 开放系统互连 文卷传送、访问和管理(idt ISO 8571:1988)

ISO 8832:1989 信息处理系统 开放系统互连 作业传送与操纵的基本类协议规范

5 定义

本定义使用在 GB/T 16505.1 中定义的数据元素、数据单元和文卷访问数据单元等术语。

5.1 二进制串; binary string

一个由零个、一个或多个八位位组组成的有序序列。

6 缩略语

FTAM 文卷传送、访问和管理

JTM 作业传送和操纵

7 文件的语义

这种文件包含一个仅含有 0 个、1 个或多个二进制串的文卷访问数据单元。这些元素的次序是有意

义的,这些二进制串的语义不由这个文件类型来定义。

这种文件结构采取如同顺序平坦约束集(见表 22)约束的 FTAM 分级文卷模型所允许的形式,这些定义出现在 GB/T 16505.2 中。

每一个二进制串由值为 0 到 255 的八位位组组成。赋与这些值的意义不受该文件类型限制。

除在这里说明的之外,这个定义没有隐含其他大小或长度限制。每个二进制串具有由 maximum-string-length 参数给出的字符数所确定的一个长度,如果 maximum-string-length 参数不存在,则缺省规则是二进制串无界。

二进制串准确的有效位是由 String-significance 参数决定的。如果它的值是“variable”,或这个参数

不存在,则这个二进制串的长度小于或等于给定的长度。如果这个值是“fixed”,则每个二进制串的长度严格地等于给定的长度。如果这个值是“not-significant”,则当文卷被存储时二进制串的边界不需要保留且对该文件的语义不起作用。

8 抽象语法结构

本文件的抽象语法结构是由在 GB/T 16505 中 ASN.1 模块 ISO 8571-FADU 和 ISO 8571-CONTENTS 中所定义的分级结构文卷,其中每一个文卷内容数据元素具有在 GB/T 16505 中文件类型登记项 FTAM-3 中定义的“ISO FTAM unstructured text”文件的抽象语法结构。

9 传送的定义

9.1 数据类型定义

本文件由具有下列任一类型的数据值组成

a) 表 22 中定义的 Datatype1;

b) 表 22 中定义的 Datatype2,它们是在 ASN.1 模块 ISO 8857-FADU 中被说明为 Data-Element 的 ASN.1 数据类型。

9.2 表示数据值

该文件作为一系列表示数据值传送,其中每个值是下列两种情况之一:

a) ASN.1 数据类型 Datatype1 的值,它携带从文件来的一个二进制串。所有值都用(任意)同一个已建立的表示上下文传输,该表示上下文支持在表 22 中说明的抽象语法名“asname1”。

b) ASN.1 数据类型 Datatype2 的值。所有值都用(任意)同一个已建立的表示上下文传输,该表示上下文支持在表 22 中说明的抽象语法名“asname2”。

注

1 特定的载体标准可能对要使用的表示上下文有附加约束,此时上述原则允许选择表示上下文。

2 在此项中定义的任何文件类型或者不使用 Datatype2,或者以 Datatype2 开始传输。

P-DATA 原语之间的边界是在传输时由发送实体在本地选择的,并且不携带文件类型的任何语义,支持这个文件类型的接收者将接受一个具有任何允许的传送选项的该类文件。

9.3 表示数据值的序列

当根据 GB/T 16505.2 中分级文卷模型定义将文卷平坦化时,类型(a)的表示数据值序列和类型(a)与(b)的表示数据值序列分别与一个数据单元中的,以及分级结构中诸数据单元的字符串序列相同。

10 传送语法

支持这些文件类型的实现对传送的所有表示数据值应支持在表 22 中命名的传送语法产生规则。

一个实现还可有选择地支持其他传送语法。

11 ASE 特定规范

11.1 GB/T 16505(FTAM)

11.1.1 简化和松弛

11.1.1.1 结构简化

此种简化丢失信息。

文件类型 FTAM-4 可被简化成文件类型 FTAM-3。结果文件含有的数据值序列与按访问上下文 UA 访问已有结构的文本文卷所得的结果相同。这就是说,只有在抽象语法“asname1”中的表示数据值才存在。

注:文件访问数据单元之间的边界保持为串之间的边界,但是任何赋予它的特殊意义都将失去。

11.1.1.2 串长度松弛

这个操作丢失在文件类型标识中的显式信息。

一个 FTAM-4 类型的文件可松弛为另一个具有较大 maximum-string-length 参数值,或没有 maximum-string-length 参数(无界二进制串)的 FTAM-4 类型的文件。

11.1.2 访问上下文选择

FTAM-4 类型的文件可在顺序平坦约束集中定义的任一访问上下文中访问。在每种情形下传送的表示数据单元就是那些从在 GB/T 16505.2 中为该访问上下文定义的结构元素中得到的数据单元。

在访问上下文 FA 和 HA 中,结果文件遵从类型 FTAM-4。在访问上下文 UA 中,结果文件遵从类型 FTAM-3。

11.1.3 插入操作

当插入操作应用在文卷尾部时,传送的东西应是一系列 FADU,这些 FADU 应以在访问上下文 FA 中用相同参数读任一 FTAM-4 类型文件的方式生成。

11.2 ISO 8832(JTM)

下列几条适用于宣称支持 FTAM-4 的任何 JTM 实现。

11.2.1 支持其他文件类型

文件类型 FTAM-3 应也被支持。

11.2.2 长度和参数

基于来自文件中规定的八位位组数的可能实现限制,实现应能够处理具有各种参数值的任何 FTAM-4 文件类型。

注:关于“支持”的规范见 ISO 8832。

实现的限制应允许使用至多包含 64 000 个八位位组的文件,这些限制应在协议实现的一致性声明中描述。

11.2.3 拼接

应支持 FTAM-4 类型文件本身的拼接,并且应产生一个包含组合的字符串序列的相同类型的文件。

注:原文件的边界在结果文件中仅看作是二个数据单元之间的常规边界。

11.2.4 求精

11.2.4.1 长度求精

由 J-GIVE 通过使用 FTAM-4 文件类型请求的任何文件应从存储的 FTAM-4 文件类型中获得,这个 FTAM-4 文件是按 11.1.1.2 中的规则处理所获得的一个松弛。这个处理可能产生一个文件,或产生一个诊断。详见 ISO 8832。

11.2.4.2 结构求精

任一由 J-GIVE 用 FTAM-4 文件类型请求的文件,应从存储的 FTAM-3 文件类型中通过把整个 FTAM-3 文件作为 FTAM-4 文件的单个数据单元(唯一的数据单元)而得到。

11.2.5 结构简化

任何由 J-GIVE 用 FTAM-3 文件类型请求的文件应通过应用在 11.1.1.1 中规定的结构简化从一个 FTAM-4 文件类型中得到。

11.2.6 松弛

对于由 J-GIVE 通过使用 FTAM-4 文件类型请求的任何文件,若该文件类型是被存储的文件类型的一个松弛,则应按所请求的“FTAM-2”文件类型提供。

注:在 J-GIVE 响应中的值从不紧缩。

B5 简单分级文卷文件类型

1 项号:FTAM-5

2 信息客体

表 23 分级文件类型中的信息客体

文件类型名	{iso standard 8571 document-type(5) simple-hierarchy(5)} “ISO FTAM simple hierarchical file”
抽象语法名	在引用文件类型时规定
传送语法名	在引用文件类型时规定
参数语法	在引用文件类型时规定
文卷模型	{iso standard 8571 file-model(3) hierarchical(1)} “FTAM hierarchical file model”
约束集	在引用文件类型时规定
文件内容	在引用文件类型时规定

3 范围

本文件类型为存储、为 FTAM 传送定义了一个带串结点标识符的简单分级文卷内容的结构。它用文件类型定义来引用,这些定义规定了此结构中数据元素的抽象语法,以出给一个完整的文卷定义。

注:这不是内容类型的有效值,且仅能被间接引用。

4 引用标准

GB/T 16505—1996 信息处理系统 开放系统互连 文卷传送、访问和管理(idt ISO 8571:1988)

5 定义

本定义使用在 GB/T 16505.1 中定义的数据元素、数据单元和文卷访问数据单元等术语。

6 缩略语

FTAM 文卷传送、访问和管理

7 文件的语义

本文件由分级文卷组成,其中的文卷访问数据单元用唯一的串标识符命名。

8 抽象语法结构

本文件的抽象语法结构是如在 GB/T 16505 中的 ASN.1 模块 ISO 8571-FADU 中定义的一个分级结构文卷,其中每一个文卷内容数据元素可有任意的抽象语法,在引用此文件类型时规定;并且在 ISO 8571-FADU 中的数据元素 Node-Name 采用选项“ftam-coded”。

9 传送的定义

传送过程的定义应在任一引用 FTAM-5 的文件类型中规定。

10 传送语法

传送语法应在任一引用 FTAM-5 的文件类型中规定。

11 ASE 特定的规范

任何 ASE 特定规范应在引用 FTAM-5 的文件类型中规定。

附 录 C

(提示的附录)

结构文卷的读

C1 引言

本附录含有在不同文卷结构中出现的的项目的内容和顺序的例子。在不同访问上下文中读取这些文卷的效果在一系列图中显示。对于每一个例子,显示文卷结构、在遍历序列中结构项目的顺序、以及在各

种不同访问上下文中传送项目的选择。在每一个例子中,假定在读操作中标识的是文卷的根结点。

文卷模型的定义、适用于它的约束集,以及读取该文卷要用的访问上下文在第 7 章中给出。在以下的描述中,第 7 章中定义的术语缩写如下:

- a) Node 代表一个 node-descriptor-data-element;
- b) DU 代表一个数据单元(Data Unit),它是一个 file-contents-data-element 序列;
- c) Enter 代表一个 enter-subtree-data-element;
- d) Exit 代表一个 exit-subtree-data-element。

在表目中的一个圆点表示这个数据元素在该访问上下文中传送。

C2 无结构文卷

一个无结构文卷由一个根节点和一个相关联的数据单元组成(见图 5)。这是由无结构约束集表达的通用分级文卷模型的一个特殊情况。

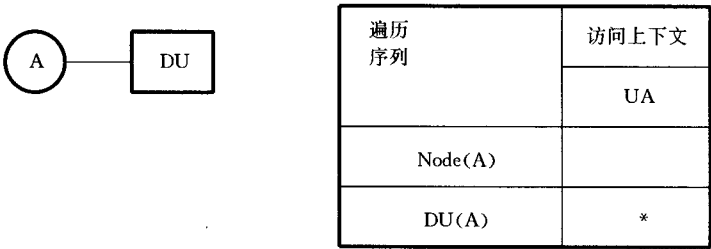


图 5 无结构文卷传送

C3 平坦文卷

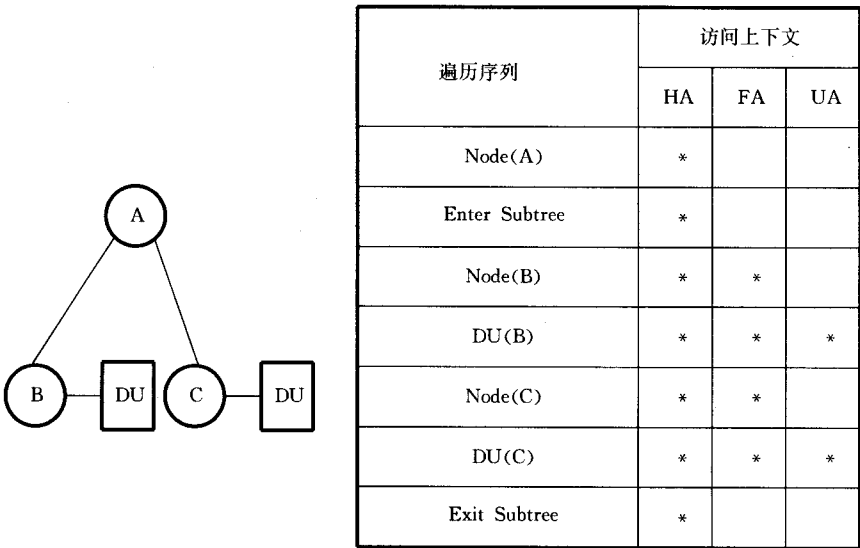


图 6 平坦文卷传送

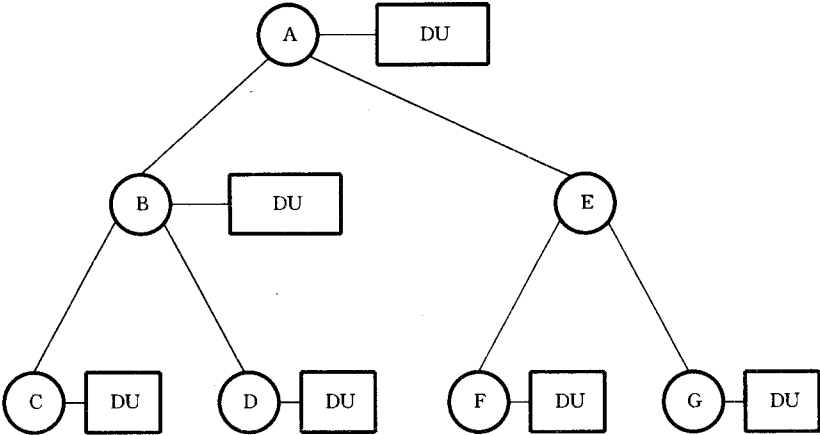
一个平坦文卷由一个根节点和一系列通过长度为 1 的弧连接到该根节点的叶节点组成。每一个叶节点带有一个数据单元(见图 6)。有三个可适用于通用分级模型的平坦约束集:顺序平坦,有序平坦,和带唯一名字的有序平坦。

在访问上下文 FA 中,不传送根节点描述符“A”,因为它不带数据单元,从而“data exists”指示符也置为假。

C4 分级文卷

有三个用不同节点标识来描述分级文卷的约束集:有序分级、通用分级,和带唯一名字的通用分级。

在结构方面，它们都产生通用分级文卷。图 7 显示了一个分级文卷的例子。



遍历序列	访问上下文						
	HA	HN	FA	FL (注)	FS	UA	US
Node(A)	*	*	*		*		
DU(A)	*		*		*	*	*
Enter Subtree	*	*					
Node(B)	*	*	*				
DU(B)	*		*			*	
Enter Subtree	*	*					
Node(C)	*	*	*	*			
DU(C)	*		*	*		*	
Node(D)	*	*	*	*			
DU(D)	*		*	*		*	
Exit Subtree	*	*					
Node(E)	*	*					
Enter Subtree	*	*					
Node(F)	*	*	*	*			
DU(F)	*		*	*		*	
Node(G)	*	*	*	*			
DU(G)	*		*	*		*	
Exit Subtree	*	*					
Exit Subtree	*	*					

注：在此例中，在上下文 FL 里的访问是在级 2。

图 7 分级文卷传送

附录 D

(提示的附录)

结构文卷中的插入

D1 例子的基础

本附录举出一系列修改结构文卷的例子,说明常规和限定动作用来对结构产生不同效果的方式。为简单起见,把动作描述成原子的,但实际上它们将被一项一项地执行。

在前五个例子中,要传送的信息从如图 8 所示的三节点文卷中得到。要修改的文卷是如图 9 中所示的一个稍复杂的文卷。

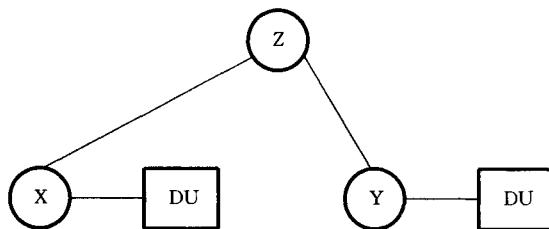


图 8 被传送数据的源

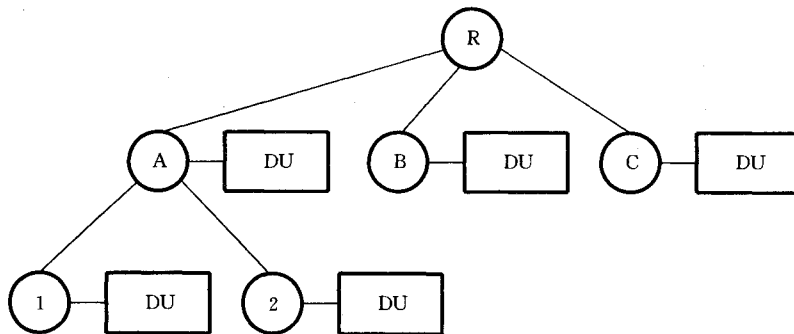


图 9 宿文卷的初始状态

D2 给 A 增加一个姐妹节点

两个来自源文卷的叶 FADU X 和 Y 要加入为 A、B 和 C 的姐妹节点。假设引用传送的 FADU 标识是“A”。传送的数据是

Node(X) DU(X) Node(Y) DU(Y)

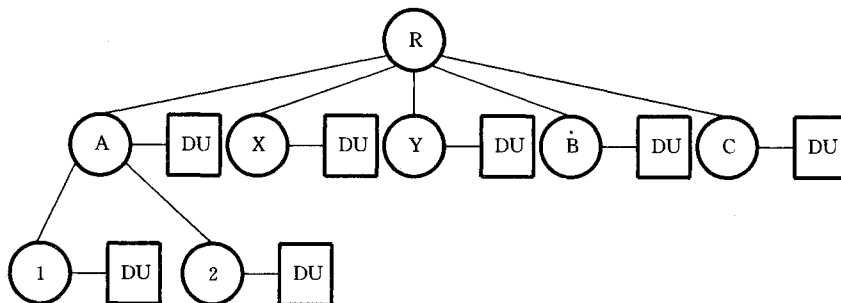


图 10 对 A 作姐妹节点插入动作的最终状态

在传送的开始时,作为引用的 FADU 标识的结果定位为“A”。接收者分析输入数据流,识别文卷结构项并动作如下:

- a) FADU“X”(Node(X)+DU(X))在遍历序列中被插入到 FADU“A”之后,“X”成为当前位置。
 b) FADU“Y”(Node(Y)+DU(Y))在遍历序列中被插入到 FADU“X”之后;“Y”成为当前位置。
 结果文卷在图 10 中显示。

D3 加入 C 的子节点(常规用法)

两个来自源文卷的叶 FADU 要加入为 C 的子节点。为传送引用的 FADU 标识是“C”。传送的数据是

Enter Node(X) DU(X) Node(Y) DU(Y) Exit

在传送的开始时,作为引用 FADU 标识的结果的定位在“C”,接收者分析输入数据流,识别文卷结构项并动作如下:

- a) 一个子节点项(Enter+Node(X)+DU(X)+Node(Y)+DU(Y)+Exit)被识别;保存当前位置并在“C”的第一个子节点前创建一临时位置(虽然实际上此时 C 没有子节点);
 b) FADU“X”插入在此位置之后。FADU“X”变成当前位置;
 c) FADU“Y”插入在此位置之后;
 d) 限定动作序列的结束是通过第一个 Exit 的出现来识别的,并将位置恢复成保存的值,“C”。
 结果结构在图 11 显示中。

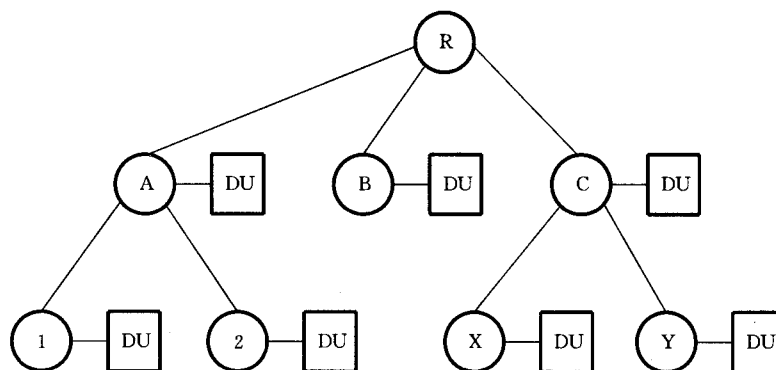


图 11 对 C 作子节点插入动作的最终状态(常规)

D4 加入 C 的子节点(变体)

此例显示了在加入子节点时中断限定序列的结果;插入的顺序被颠倒。这么做的实际用途还不清楚,引用这个例子是为了显示插入过程。两个来自源文卷的叶 FADU X 和 Y 要加入作为 C 的子节点。为传送而引用的 FADU 标识是“C”。传送的数据是

Enter Node(X) DU(X) Exit Enter Node(Y) DU(Y) Exit

在传送的开始时,作为引用 FADU 标识的结果定位为“C”。接收者分析输入数据流,识别文卷结构项并动作如下:

- a) 一个子节点项(Enter+Node(X)+DU(X)+Exit)被识别;保存当前位置并在“C”的第一个子节点前创建一临时位置(虽然实际上此时 C 没有子节点);FADU“X”插入在此位置之后。简而言之, FADU“X”成为当前位置,但一旦限定动作序列的结束通过第一个 Exit 的出现被识别,则位置被恢复成保存的值,“C”。
 b) 第二个子节点项(Enter+Node(Y)+DU(Y)+Exit)被识别;再次保存当前位置并在“C”的第一个子节点前创建一临时位置(这时在“X”之前);FADU“Y”插入在此位置之后。限定动作序列的结束又通过 Exit 的出现被识别,于是位置再被恢复成保存的值,“C”。

结果结构在图 12 中显示。

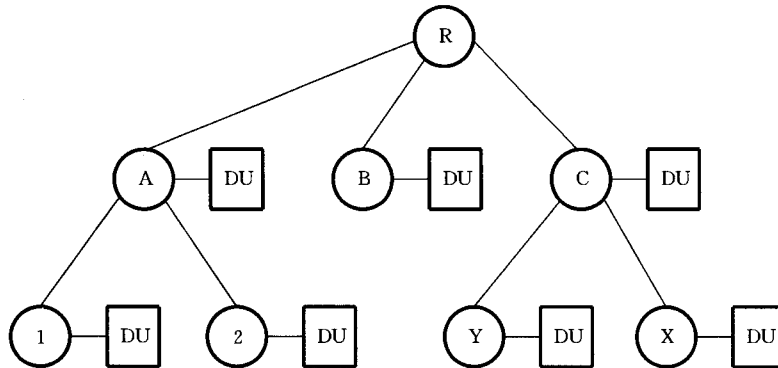


图 12 对 C 作子节点插入动作的最终状态(变体)

D5 插入子树作为姐妹节点

整个源子树加入为节点 A、B 和 C 的一个姐妹节点。假定为传送而引用的 FADU 标识是“C”。传送的数据是

Node(Z) Enter Node(X) DU(X) Node(Y) DU(Y) Exit

在传送的开始时,作为引用的 FADU 标识的结果定位为“C”。接收者分析输入数据流并且识别出整个数据流是根在 Z 的一个子树编码,因此将此子树表示的 FADU 作为 A、B 和 C 的姐妹节点插入文卷中,按遍历序列跟在 C 之后。“Z”成为当前位置。

结果文卷在图 13 中显示。

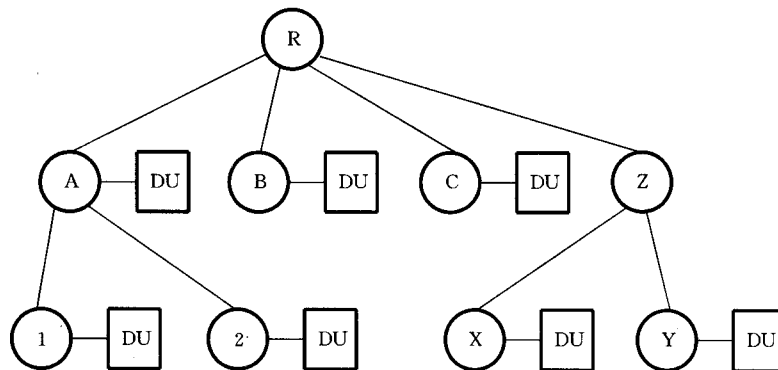


图 13 插入子树作为姐妹节点的最终状态

D6 插入子树作为子节点

整个子树插入作为 C 的子节点。为传送引用的 FADU 标识是“C”。传送的数据是

Enter Node(Z) Enter Node(X) DU(X) Node(Y) DU(Y) Exit Exit

在传送的开始时,作为引用 FADU 标识的结果定位为“C”。接收者分析输入数据流并且识别出整个数据流是一个仅含一根为 Z 的子树的“子节点”项。该位置被保存且在“C”的第一个子节点前创建一个临时位置(虽然实际上此时“C”没有子节点);子树插入在此位置后。简而言之,子树“Z”的根成为当前位置,但一旦限定动作序列的结尾通过第一个 Exit 的出现被识别,位置恢复为保存值,“C”。

文卷的结果状态在图 14 中显示。

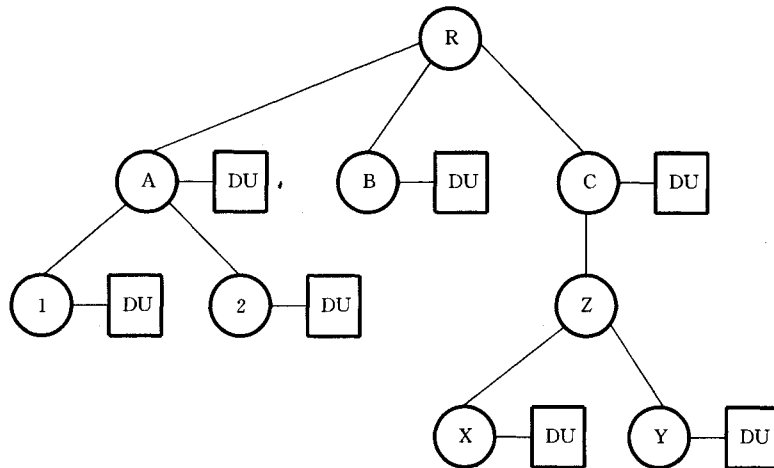


图 14 插入子树作为子节点的最终状态

D7 在一个有序平坦文卷中插入

最后一个例子显示将指定叶节点合并到一有序平坦文卷。在这个例子中，目的文卷的叶节点标识符假定按字母顺序排列。文卷的初始状在图 15 中显示。

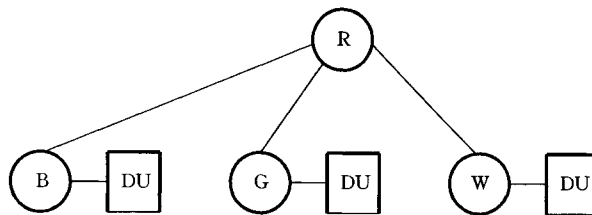


图 15 有序平坦文卷的初始状态

带“A”、“P”和“X”标识符的节点将合并到该文卷。传送的数据是
Node(X) DU(X) Node(A) DU(A) Node(P) DU(P)

在传送的开始时，作为引用 FADU 标识的结果定位在根节点。接收者分析输入数据流，识别文卷结构项并动作如下：

- FADU“X”(Node(X)+DU(X))按分类顺序插在现存的 FADU“W”之后。
- FADU“A”(Node(A)+DU(A))按分类顺序插在现存的 FADU“B”之前。
- FADU“P”(Node(P)+DU(P))按分类顺序插在现存的 FADU“G”之后。

结果文卷在图 16 中显示。

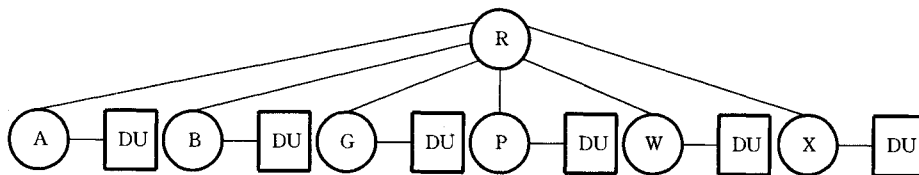


图 16 合并有序平坦文卷的最终状态

附录 E

(提示的附录)

ASN.1 交叉对照表

E1 在 ISO 8571-CONTENS 模块中类型定义/引用的交叉对照

AnyType	引用行:4
File-Contents-Data-Element	定义行:4
	未被引用

E2 在 ISO 8571-FADU 模块中类型定义/引用的交叉对照

BooleanType	引用行:27([2] IMPLICIT)
Children	定义行:12
	引用行:9([1] IMPLICIT)
ChoiceType	引用行:30
	行:49
DU	定义行:18
	引用行:7([0] IMPLICIT)
Enter-Subtree-Data-Element	定义行:41
	引用行:13
	行:51
Exit-Subtree-Data-Element	定义行:43
	引用行:16
	行:52
External Type	引用行:37
FADU	定义行:47
	未被引用
GraphicStringType	引用行:31([0] IMPLICIT)
IntegerType	引用行:25([1] IMPLICIT)
Node-Descriptor-Data-Element	定义行:21
	引用行:6
	行:50
Mode-Name	定义行:30
	引用行:23
NullType	引用行:41([APPLICATION 1] IMPLICIT)
	行:43([APPLICATION 2] IMPLICIT)
SequenceOfType	引用行:14
	行:18
SequenceType	引用行:5
	行:12
	行:22([APPLICATION 0] IMPLICIT)
Structuring-Data-Element	定义行:49

Subtree	未被引用 定义行:5 引用行:14 行:47
E2.1 在模块 ISO 8571-CONTENTS 中定义的类型引用	
File-Contents-Data-Element	引用行:18



GB/T 16505.2-1996

版权专有 不得翻印

*

书号:155066·1-14151

定价: 23.00 元

*

标 目 321—46