



# 中华人民共和国国家标准

GB/T 16264.5—2008/ISO/IEC 9594-5:2005  
代替 GB/T 16264.5—1996

---

## 信息技术 开放系统互连 目录 第 5 部分:协议规范

**Information technology—Open Systems Interconnection—The Directory—  
Part 5: Protocol specifications**

(ISO/IEC 9594-5:2005 Information technology—Open Systems  
Interconnection—The Directory: Protocol specifications, IDT)

2008-08-06 发布

2009-01-01 实施

---

中华人民共和国国家质量监督检验检疫总局 发布  
中国国家标准化管理委员会

国家图书馆专用

## 目 次

前言 .....	Ⅲ
引言 .....	Ⅳ
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	2
4 缩略语 .....	4
5 约定 .....	5
6 公共协议规范 .....	5
7 使用 OSI 栈的目录协议 .....	9
8 目录协议映射到 OSI 服务 .....	27
9 IDM 协议 .....	36
10 目录协议映射到 IDM 协议 .....	43
11 协议栈共存 .....	44
12 版本及扩展规则 .....	45
13 一致性 .....	48
附录 A (规范性附录) 用 ASN.1 描述的公共协议规范 .....	54
附录 B (规范性附录) 用 ASN.1 描述的 OSI 协议 .....	57
附录 C (规范性附录) 用 ASN.1 描述的目录 OSI 协议 .....	66
附录 D (规范性附录) 用 ASN.1 描述 IDM 协议 .....	70
附录 E (规范性附录) 用 ASN.1 描述的目录 IDM 协议 .....	73
附录 F (规范性附录) 目录操作绑定类型 .....	75

国家图书馆专用

## 前 言

GB/T 16264《信息技术 开放系统互连 目录》包括以下 10 个部分：

- 第 1 部分：概念、模型和服务的概述；
- 第 2 部分：模型；
- 第 3 部分：抽象服务定义；
- 第 4 部分：分布式操作规程；
- 第 5 部分：协议规范；
- 第 6 部分：选定的属性类型；
- 第 7 部分：选定的客体类；
- 第 8 部分：公钥和属性证书框架；
- 第 9 部分：复制(待发布)；
- 第 10 部分：公用目录管理机构的系统管理用法(待发布)。

本部分是 GB/T 16264 的第 5 部分。

本部分等同采用 ISO/IEC 9594-5:2005《信息技术 开放系统互连 目录 协议规范》，仅有编辑性修改。

本部分代替 GB/T 16264.5—1996。

本部分与 GB/T 16264.5—1996 的差异如下：

- 增加了对 TCP/IP 协议的支持；
- 增加了对 OSI 和 TCP/IP 两种协议共存的支持；
- 增加了互联网直接映射；
- 增加了版本及扩展规则。

本部分的附录 A、附录 B、附录 C、附录 D、附录 E、附录 F 是规范性附录。

本部分由中华人民共和国信息产业部提出。

本部分由全国信息技术标准化技术委员会归口。

本部分起草单位：中国电子技术标准化研究所。

本部分主要起草人：徐冬梅、郑洪仁、郭楠、胡顺。

本部分于 1996 年首次发布，本次为第一次修订。

## 引 言

GB/T 16264 的本部分连同本标准其他部分是方便信息处理系统之间的互连以提供目录服务而制定的。所有这些系统的集合,连同它们所拥有的目录信息可被视为一个整体,被称为“目录”。目录所拥有的信息,总称为目录信息库(DIB),典型地被用于方便客体之间的通信、与客体的通信或有关客体的通信等,这些客体如应用实体、个人、终端和分布列表等。

目录在开放系统互连中扮演了重要角色,其目标是,在它们自身的互连标准之外做最少的技术约定的情况下,允许下述各种信息处理系统之间的互连:

- 来自不同生产厂商;
- 具有不同的管理;
- 具有不同的复杂程度,以及
- 有不同的年代。

本部分规定了两个协议——目录访问协议(DAP)和目录系统协议(DSP)的应用服务元素和应用上下文。DAP 供访问目录之用,以便检索或修改目录信息。DSP 供链接请求之用,以便对可以拥有相应信息的分布式目录系统的其他部分进行目录信息的检索或修改。

此外,本部分还规定了目录信息影像协议(DISP)和目录操作绑定管理协议(DOP)的应用服务元素和应用上下文。DISP 将某个 DSA 中拥有的信息影像到另一个 DSA。DOP 在 DSA 对之间提供绑定的建立、修改和终止功能,以便管理 DSA 之间的关系(例如影像关系或等级关系)。

本部分提供了一个基础框架,在此框架基础上,其他标准化组织和业界论坛可以定义工业配置集。在本框架中定义为可选的许多特性,可通过配置集的说明,在某种环境下作为必选特性来使用。ISO/IEC 9594 的第 5 版是原有国际标准第 4 版的修订和增强,但不是替代。在系统实现时仍可以声明为遵循第 4 版。然而,在某些方面,将不再支持第 4 版(即不再消除一些报告上来的差错)。建议在系统实现时尽快遵循第 5 版。

第 5 版详细定义了目录协议的第 1 版和第 2 版。

第 1 版和第 2 版仅定义了协议第 1 版。本版本(第 5 版)中定义的许多服务和协议被设计为可运行在第 1 版下。然而,一些增强的服务和协议,如署名差错,只有包含在操作中的所有的目录条目都协商支持协议第 2 版时才可运行。无论协商的是哪一版,第 5 版中所定义的服务之间的差异和协议之间的差异,除了那些特别分配给第 2 版的外,都可以使用 GB/T 16264.5—2008 中定义的扩展规则调节。

本部分使用术语“第 1 版系统”来指遵循国际标准第 1 版的所有系统,即 ISO/IEC 9594:1990 版本;本部分使用术语“第 2 版系统”来指遵循国际标准第 2 版本的所有系统,即 ISO/IEC 9594:1995 版本;本部分使用术语“第 3 版系统”来指遵循国际标准第 3 版的所有系统,即 ISO/IEC 9594:1998 版本;本部分使用术语“第 4 版系统”来指遵循国际标准第 4 版的所有系统,即 ISO/IEC 9594:2001 版本的第 1 部分到第 10 部分;本部分使用术语“第 5 版系统”来指遵循国际标准第 5 版的所有系统,即 ISO/IEC 9594:2005 版本。

GB/T 16264—1996 是参照 ISO/IEC 9594:1990 而制定的。我国没有制定与国际标准第 2 版、第 3 版、第 4 版对应的国家标准。本部分提到的版本号是指国际标准的版本号。

附录 A 是规范性附录,提供了目录协议公共规范的 ASN.1 模块定义。

附录 B 是规范性附录,提供了 OSI 协议规范的 ASN.1 模块定义。

附录 C 是规范性附录,提供了目录 OSI 协议的 ASN.1 模块定义。

附录 D 是规范性附录,提供了 IDM 协议规范的 ASN.1 模块定义。

附录 E 是规范性附录,提供了目录 IDM 协议的 ASN.1 模块定义。

附录 F 是规范性附录,提供了 ASN.1 模块定义,包含了为标识本系列部分中的操作绑定类型而分配的所有的 ASN.1 客体标识符。

国家图书馆专用

国家图书馆专用



## 信息技术 开放系统互连 目录

### 第5部分:协议规范

#### 1 范围

GB/T 16264 的本部分规定了目录访问协议、目录系统协议、目录信息影像协议和目录操作绑定管理协议,用以实现在 GB/T 16264.2—2008、GB/T 16264.3—2008、GB/T 16264.4—2008 以及 ISO/IEC 9594-9:2005 中规定的抽象服务。

#### 2 规范性引用文件

下列文件中的条款通过 GB/T 16264 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 9387.1—1998 信息技术 开放系统互连 基本参考模型 第1部分:基本模型(idt ISO/IEC 7498-1:1994)

GB/T 12453—2008 信息技术 开放系统互连 运输服务定义(ISO/IEC 8072:1996,IDT)

GB/T 15126—2008 信息技术 开放系统互连 网络服务定义(ISO/IEC 8348:2002,IDT)

GB/T 16262.1—2006 信息技术 抽象语法记法一(ASN.1) 第1部分:基本记法规范(ISO/IEC 8824-1:2002,IDT)

GB/T 16262.2—2006 信息技术 抽象语法记法一(ASN.1) 第2部分:信息客体规范(ISO/IEC 8824-2:2002,IDT)

GB/T 16262.3—2006 信息技术 抽象语法记法一(ASN.1) 第3部分:约束规范(ISO/IEC 8824-3:2002,IDT)

GB/T 16262.4—2006 信息技术 抽象语法记法一(ASN.1) 第4部分:ASN.1 规范的参数化(ISO/IEC 8824-4:2002,IDT)

GB/T 16263.1—2006 信息技术 ASN.1 编码规则 第1部分:基本编码规则(BER)、正则编码规则(CER)和非典型编码规则(DER)规范(ISO/IEC 8825-1:2002,IDT)

GB/T 16263.2—2006 信息技术 ASN.1 编码规则 第2部分:紧缩编码规则(PER)规范(ISO/IEC 8825-2:2002,IDT)

GB/T 16264.1—2008 信息技术 开放系统互连 目录 第1部分:概念、模型和服务的概述(ISO/IEC 9594-1:2005,IDT)

GB/T 16264.2—2008 信息技术 开放系统互连 目录 第2部分:模型(ISO/IEC 9594-2:2005,IDT)

GB/T 16264.3—2008 信息技术 开放系统互连 目录 第3部分:抽象服务定义(ISO/IEC 9594-3:2005,IDT)

GB/T 16264.4—2008 信息技术 开放系统互连 目录 第4部分:分布式操作规程(ISO/IEC 9594-4:2005,IDT)

GB/T 16264.6—2008 信息技术 开放系统互连 目录 第6部分:选定的属性类型(ISO/IEC 9594-6:2005,IDT)

## GB/T 16264.5—2008/ISO/IEC 9594-5:2005

GB/T 16264.7—2008 信息技术 开放系统互连 目录 第7部分:选定的客体类(idt ISO/IEC 9594-7:2005,IDT)

GB/T 16688 信息处理系统 开放系统互连 联系控制服务元素服务定义(GB/T 16688—1996, idt ISO/IEC 8649:1988)

GB/T 16975.1—2000 信息技术 远程操作 第1部分:概念、模型和表示法(idt ISO/IEC 13712-1:1995)

GB/T 16975.2—1997 信息技术 远程操作 第2部分:OSI实现 远程操作服务元素(ROSE)服务定义(idt ISO/IEC 13712-2:1995)

GB/T 17965—2000 信息技术 开放系统互连 高层安全模型(idt ISO/IEC 10745:1995)

GB/T 18237.1—2000 信息技术 开放系统互连 通用高层安全 第1部分:概述、模型和记法(idt ISO/IEC 11586-1:1996)

GB/T 18794.2—2002 信息技术 开放系统互连 开放系统安全框架 第2部分:鉴别框架(idt ISO/IEC 10181-2:1996)

GB/T 18794.3—2002 信息技术 开放系统互连 开放系统安全框架 第3部分:访问控制框架(idt ISO/IEC 10181-3:1996)

ISO/IEC 8327-1:1996 信息技术 开放系统互连 面向连接的会话协议:协议规范

ISO/IEC 8650-1:1996 信息处理系统 开放系统互连 联系控制服务元素协议规范

ISO/IEC 8823-1:1994 信息技术 开放系统互连 面向连接的表示协议:协议规范

ISO/IEC 8825-4 信息技术 ASN.1 编码规则:XML 编码规则(XER)

ISO/IEC 9594-8:2005 信息技术 开放系统互连 目录:公钥和属性证书框架

ISO/IEC 9594-9:2005 信息技术 开放系统互连 目录:复制

ISO/IEC 9594-10:2005 信息技术 开放系统互连 目录:公用目录管理机构的系统管理用法

ISO/IEC 10646:2003 通用多八位编码字符集(UCS)

ITU-T 建议 E.164:2005 国际公共远程通信编号计划

ITU-T 建议 X.121:2000 公共数据网的国际编号方案

IETF RFC 793:1981 运输控制协议 DARPA 网际协议 DARPA 网际程序 协议规范

IETF RFC 1277:1991 在非 OSI 较低层支持操作的网络地址编码

IETF RFC 1738:1994 统一资源地址(URL)

IETF RFC 2025:1996 简单公钥 GSS-API 机制(SPKM)

IETF RFC 2246:1999 TLS 协议版本 1.0

IETF RFC 2251:1997 轻量级目录访问协议(v3)

IETF RFC 3546:2003 运输层安全性(TLS)扩展

### 3 术语和定义

下列术语和定义适用于 GB/T 16264 的本部分。

#### 3.1 基本目录定义

下列术语在 GB/T 16264.2—2008 中规定:

- a) 目录 the directory;
- b) (目录)用户 (directory) user;
- c) 目录系统代理 directory system agent(DSA);
- d) 目录用户代理 directory user agent(DSA)。

#### 3.2 分布式操作定义

下列术语在 GB/T 16264.4—2008 中规定:

- a) 链接 chaining；
- b) 转向推荐 referral。

### 3.3 协议规范定义

下列术语在 GB/T 16264 的本部分中规定。

注：本条定义的术语是通用定义，可适用于 OSI 和 TCP/IP 环境，除非有明确说明。

#### 3.3.1

##### **抽象语法 abstract syntax**

使用独立于编码技术的记法规则来表示的数据类型和/或数据值的规范。

#### 3.3.2

##### **应用联系 application-association**

通过绑定操作建立的两个应用实体之间的一种协作关系。

#### 3.3.3

##### **应用上下文 application-context**

(仅适用于 OSI)两个应用实体为了支持某个应用联系而共享的规则集。

#### 3.3.4

##### **应用上下文名称 application-context-name**

标识(或命名)某个应用上下文的一个 ASN.1 客体标识符。

#### 3.3.5

##### **应用层 Application Layer**

OSI 七层模型中的顶层，用以表示通信语义。

#### 3.3.6

##### **应用实体 application-entity**

某个应用进程外部行为的一种表示，以通信能力的形式表示。

#### 3.3.7

##### **应用实体标题 application-entity title**

某个应用实体，尤其是一个表示目录应用进程的应用实体的目录可辨别名。

#### 3.3.8

##### **应用进程 application process**

系统内的一个进程，该进程出于某种特定的目的而执行信息处理，尤其是处理目录操作。

#### 3.3.9

##### **绑定操作 Bind operation**

一种用于建立一个应用联系的操作类型。

#### 3.3.10

##### **目录操作 Directory operation**

一种用于交换目录信息的操作类型。

#### 3.3.11

##### **目录协议数据单元 directory protocol-data-unit**

某种目录协议的数据单元，包括控制信息，一般情况下还包括目录操作所规定的应用数据。

注 1：OSI 环境中的一个目录 PDU 包含 OSI 表示层的所有协议元素，并且如果相关的话，除了目录特定的协议元素外还包含 ACSE 协议元素。

注 2：术语“应用协议数据单元(APDU)”是 OSI 应用协议定义的一种数据单元。该术语不用于本系列目录规范的第 5 版本和后续版本。然而，该缩略语可以出现在某些 ASN.1 元素中。

### 3.3.12

#### 发起者 initiator

通过发出一个绑定请求而发起一个应用联系的应用进程。

### 3.3.13

#### 操作 operation

两个应用进程之间为了执行一个特定任务而进行的交换。它包括从一个应用进程向另一个应用进程的请求,以及零个或多个响应(结果和/或差错)的返回。一个操作意味着接收请求的应用进程将执行某种处理。

### 3.3.14

#### 协议数据单元 protocol-data-unit

由一个目录协议数据单元的代表协议元素或 ACSE 协议元素组成。

### 3.3.15

#### 表示层 Presentation Layer

OSI 参考模型的第 6 层。

### 3.3.16

#### 协议差错 protocol error

表示接收到的未识别的或非期望的协议数据单元,或者带有非期望的或无效参数的协议数据单元。

### 3.3.17

#### 响应者 responder

接收到一个绑定请求的应用进程,该进程或者接受或者拒绝此应用联系。

### 3.3.18

#### 会话层 session layer

OSI 参考模型的第 5 层。

### 3.3.19

#### 会话协议数据单元 session-protocol-data-unit

(仅适用于 OSI) OSI 会话层的一个数据单元,包括控制信息,一般情况下还携带目录协议数据单元。

## 4 缩略语

下列缩略语适用于 GB/T 16264 的本部分:

AC	应用上下文	(Application Context)
ACSE	联系控制服务元素	(Association Control Service Element)
AE	应用实体	(application-entity)
APDU	应用协议数据单元	(application-protocol-data-unit)
DAP	目录访问协议	(Directory Access Protocol)
DISP	目录信息影像协议	(Directory Information Shadowing Protocol)
DOP	目录操作绑定管理协议	(Directory Operational Binding Management)
DSA	目录系统代理	(Directory System Agent)
DSP	目录系统协议	(Directory System Protocol)
DUA	目录用户代理	(Directory User Agent)
IDM	互联网直接映射	(Internet Directly Mapped)
LDAP	轻量级目录访问协议	(Lightweight Directory Access Protocol)
PDU	协议数据单元	(protocol-data-unit)

PPDU	表示协议数据单元	(presentation-protocol-data-unit)
SPDU	会话协议数据单元	(session-protocol-data-unit)
TCP/IP	传输控制协议/网际协议	(Transmission Control Protocol/Internet Protocol)
TSDU	运输服务数据单元	(transport-service-data-unit)

## 5 约定

术语“目录规范(或本目录规范)”指的是 GB/T 16264.5。术语“系列目录规范”指的是 GB/T 16264 (或者 ISO/IEC 9594)的所有部分。

本目录规范使用术语“第 1 版系统”来指遵循系列目录规范第 1 版的所有系统,即 GB/T 16264—1996 版本。本目录规范使用术语“第 2 版系统”来指遵循系列目录规范第 2 版本的所有系统,即 ISO/IEC 9594:1995 版本。本目录规范使用术语“第 3 版系统”来指遵循系列目录规范第 3 版的所有系统,即 ISO/IEC 9594:1998 版本。本目录规范使用术语“第 4 版系统”来指遵循系列目录规范第 4 版的所有系统,即 2001 年版本的 ISO/IEC 9594:2001 年版本的第 1 到第 10 部分。

本目录规范使用术语“第 5 版系统”来指遵循系列目录规范第 5 版的所有系统,即 GB/T 16264—2008 版本的第 1 到第 7 部分以及 ISO/IEC 9594-8:2005、ISO/IEC 9594-9:2005 和 ISO/IEC 9594-10:2005。

本目录规范使用粗体字来表示 ASN.1 记法。若在常规文本中要表示 ASN.1 的类型和值时,为了区别于常规文本,使用了粗体字表示。为了表示过程的语义而引用过程名时,为了区别于常规文本,使用了粗体字表示。访问控制许可使用斜体字表示。

## 6 公共协议规范

### 6.1 目录联系和操作

本系列目录规范中的协议被描述为一系列的操作。操作被定义为从一个系统向另一个系统所发送的请求,并且期望后者来执行此请求,如果后者可应用,则返回一个或多个构成结果的答复。一个操作可以是一个绑定操作,或者是一个被调用来访问目录信息的操作(即一个目录操作)。

如果遇到异常条件,则可返回一个或多个差错,以代替或附加于可能的结果。

注 1: 当前定义的操作将返回一个或多个结果或单个差错。

本系列目录规范中定义的目录协议可以使用 OSI 协议栈、TCP/IP 协议栈或者两种协议栈均使用。本章提供的规范独立于特定的协议栈。OSI 特定的规范在第 7 章和第 8 章提供,而 TCP/IP 特定的规范在第 9 章和第 10 章提供。

系统内处理目录操作的进程被称为应用进程。应用实体是某个应用进程外部行为的一种体现。

在两个目录应用进程之间发起目录操作调用之前,须在相应的两个应用实体间首先建立应用联系。应用联系是两个应用实体之间的一种协作关系,其建立是通过在一个绑定操作的请求和结果中交换控制信息,并且通过使用一个共同的底层服务而完成的。

注 2: 这是对 GB/T 16688 中定义的应用联系的一种修改定义,以备既可适用于底层 OSI 协议栈也可用于底层 TCP/IP 协议栈。

一个应用联系通过一个解绑定的交换来终止。一个应用联系的解绑定没有被定义为一种操作。

### 6.2 目录操作规范

本系列目录规范定义了多种操作类型。操作类型通过 ASN.1 的 OPERATION 信息客体类来规定。与某种操作类型联系的可能的差错通过 ASN.1 的 ERRORS 信息客体类来定义。

```
OPERATION ::= CLASS {
    &.ArgumentType,
    &.ResultType OPTIONAL,
```

```

    &.Errors      ERROR OPTIONAL,
    &.operationCode Code UNIQUE OPTIONAL}
WITH SYNTAX{
    ARGUMENT    &.ArgumentType
    [RESULT      &.ResultType]
    [ERRORS      &.Errors]
    [CODE        &.operationCode]}

```

```

ERROR ::= CLASS {
    &. ParameterType,
    &.errorCode Code UNIQUE OPTIONAL}

```

```

WITH SYNTAX {
    PARAMETER &. ParameterType
    [CODE      &.errorCode]}

```

```

Code ::= CHOICE {
    local      INTEGER,
    global     OBJECT IDENTIFIER }

```

信息客体类OPERATION 是一种表示某个特定操作类型的目录请求、结果和差错的语法的便捷方法。

此 ASN.1 信息客体类具有下列字段：

- 字段&.ArgumentType 为一个操作的请求部分规定了一种开放的数据类型；
- 字段&.ResultType 为组成请求结果的一个或多个答复规定了一种开放的数据类型。如果该字段不存在，则此操作没有相关的结果；
- 字段&.Errors 规定了处理请求的结果可以出现的一个或多个差错。如果该字段不存在，则此操作没有相关的差错；
- 字段&.operationCode 规定了将要执行的目录操作的类型。对于绑定操作，此字段不存在。目前定义的操作代码见 6.4。

目录操作大致可通过两种不同的方式执行：

- 如果一个目录操作应在发起另一个新的目录操作之前完成，则操作方式为同步；或者
- 如果多个操作可以在同一时间进行，则操作方式为异步。

如果为某种特定类型的应用联系所定义的所有目录操作：

- 既包含一个请求，又包含一个或多个结果和/或差错；以及
- 仅允许被一个指定的系统所调用。

则这样的操作可以同步或异步方式执行。否则，操作的方式总是异步的。

信息客体类OPERATION 本身并不隐含任何顺序。某个目录请求可以没有结果和/或差错，或者某个请求可具有多个结果和/或差错。它是通过携带相同的操作代码和相同的调用 ID(见下)，将一个请求与可能的响应(结果和差错)系在一起。然而，某种特定操作类型的规范也可规定顺序上的限制。

差错是对某个操作不成功执行的一种报告。差错由ERROR 的 ASN.1 信息客体类来表示。不同的字段描述如下：

- 字段&.ParameterType 规定了差错参数的数据类型，这些参数规定了差错的性质；
- 字段&.errorCode 规定了标识差错的代码(所定义的差错代码见 6.5)。

尽管没有体现在信息客体类 OPERATION 或 ERRORS 中, 但一个目录操作的每次调用都被分配了一个 InvokeId, 该调用标识符携带在协议中。这使得指明一个特定的请求、结果或差错属于哪个目录操作成为可能。

InvokeId 的定义如下所述:

```
InvokeId ::= CHOICE {
    present INTEGER,
    absent NULL}
```

如果操作类型没有规定 &operationCode, 则这种类型的操作不能被分配 InvokeId。

### 6.3 目录协议概述

#### 6.3.1 底层服务的使用

当来自不同开放系统的两个应用进程交互时, 应用联系被由或者使用 OSI 或者使用 TCP/IP 底层服务的应用层协议来实现。

使用 OSI 服务的详细信息见第 8 章, 而使用 TCP/IP 服务的详细信息见第 10 章。

#### 6.3.2 目录访问协议(DAP)

当来自不同开放系统的 DUA 和 DSA 能进行交互之前, 应在它们之间调用绑定操作来建立应用联系, 用以支持目录协议, 此目录协议被称为目录访问协议(DAP)。

建立一个 DAP 应用联系的绑定操作(directoryBind)在 GB/T 16264.3—2008 的第 8 章中定义。

本版本以及本系列目录规范的所有之前版本都仅允许一个 DUA 来调用一个绑定操作, 并且发起后续的目录操作。如果使用的是 OSI 底层协议栈, 则目录操作可以以同步方式或异步方式被调用。如果使用的是 TCP/IP 底层协议栈, 则目录操作将总是以异步方式被调用。

所有的目录操作都要求返回单个答复或单个差错。

#### 6.3.3 目录系统协议(DSP)

当来自不同开放系统的一对 DSA 进行交互之前, 应在它们之间调用绑定操作来建立应用联系, 用以支持目录协议, 此目录协议被称为目录系统协议(DSP)。

建立 DSP 应用联系的绑定操作(dSABind)在 GB/T 16264.4—2008 的第 11 章中定义。

任意 DSA 都可以调用绑定操作。发起方 DSA 和响应方 DSA 都可以调用后续的目录操作。DSP 上的目录操作将总是以异步方式被调用。

所有的目录操作都要求返回单个答复或单个差错。

#### 6.3.4 目录信息影像协议(DISP)

当来自不同开放系统的一对 DSA 出于交换影像信息的目的进行交互之前, 须在它们之间调用绑定操作来建立应用联系, 用以支持目录协议, 此目录协议被称为目录信息影像协议(DISP)。

建立一个 DISP 应用联系的绑定操作(dSAShadowBind)在 ISO/IEC 9594-9:2005 的 7.4.1 中定义。

如果使用的是 OSI 底层协议栈, 则操作模式是同步方式或异步方式依赖于为绑定操作所选择的应用上下文。如果使用的是 TCP/IP 底层协议栈, 则目录操作将总是以异步方式被调用。

所有的目录操作都要求返回单个答复或单个差错。

#### 6.3.5 目录操作绑定管理协议(DOP)

当来自不同开放系统的一对 DSA 出于维护操作绑定的目的进行交互之前, 应调用绑定操作来建立应用联系, 用以支持目录协议, 此目录协议被称为目录操作绑定管理协议(DOP)。

可以在绑定操作中担任发起者角色的 DSA, 依赖于在应用联系中使用目录操作为被管理的操作绑定所分配的 DSA 角色。仅有发起者可以发起目录操作调用。只有当不同类型的 DSA 角色是兼容的(例如, 一个 DSA 为每种绑定类型都担任角色 A)时, 才可以有多于一个的操作绑定类型在此应用联系内被管理。

所有的目录操作都要求返回单个答复或单个差错。

## 6.4 操作代码

### 6.4.1 DAP 和 DSP 的操作代码

下列操作代码用于 DAP 和 DSP 中。

id-opcode-read	Code	::= local: 1
id-opcode-compare	Code	::= local: 2
id-opcode-abandon	Code	::= local: 3
id-opcode-list	Code	::= local: 4
id-opcode-search	Code	::= local: 5
id-opcode-addEntry	Code	::= local: 6
id-opcode-removeEntry	Code	::= local: 7
id-opcode-modifyEntry	Code	::= local: 8
id-opcode-modifyDN	Code	::= local: 9

这些操作代码的用法在 GB/T 16264.3—2008 中规定。

### 6.4.2 DISP 的操作代码

下列操作代码用于 DISP 中。

id-opcode-requestShadowUpdate	Code	::= local: 1
id-opcode-updateShadow	Code	::= local: 2
id-opcode-coordinateShadowUpdate	Code	::= local: 3

这些操作代码的用法在 ISO/IEC 9594-9:2005 中规定。

### 6.4.3 DOP 的操作代码

下列操作代码用于 DOP 中。

id-op-establishOperationalBinding	Code	::= local: 100
id-op-modifyOperationalBinding	Code	::= local: 102
id-op-terminateOperationalBinding	Code	::= local: 101

这些操作代码的用法在 GB/T 16264.2—2008 中规定。

## 6.5 差错代码

### 6.5.1 DAP 和 DSP 的差错代码

下列差错代码用于 DAP 和 DSP 中。代码 id-errcode-referral 仅用于 DAP 中。代码 id-opcode-dsaReferral 仅用于 DSP 中：

id-errcode-attributeError	Code	::= local: 1
id-errcode-nameError	Code	::= local: 2
id-errcode-serviceError	Code	::= local: 3
id-errcode-referral	Code	::= local: 4
id-errcode-abandoned	Code	::= local: 5
id-errcode-securityError	Code	::= local: 6
id-errcode-abandonFailed	Code	::= local: 7
id-errcode-updateError	Code	::= local: 8
id-errcode-dsaReferral	Code	::= local: 9

### 6.5.2 DISP 的差错代码

下列差错代码用于 DISP 中：

id-errcode-shadowError	Code	::= local: 1
------------------------	------	--------------



### 6.5.3 DOP 的差错代码

下列差错代码用于 DOP 中：

id-err-operationalBindingError Code ::= local: 100

### 6.6 抽象语法

协议规范包括数据类型的规范, 这些数据类型可作为协议交换的一部分被传递。数据类型使用一种类似于 ASN.1 记法的抽象记法来定义, 并由此组成协议的抽象语法。抽象语法对于 OSI 通信和 TCP/IP 通信是非常类似的, 尽管还是有所区别。相对应四种不同的目录协议, 为这些通信类型分别定义了四种抽象语法。抽象语法仅对 OSI 通信分配客体标识符。当建立 OSI 应用联系时, 抽象语法的相关客体标识符在绑定中被告知(见 7.6.1)。

## 7 使用 OSI 栈的目录协议

本章定义了目录协议, 以及它们如何映射到 OSI 的会话协议。它将 ISO/IEC 8823-1 中定义的 OSI 表示协议的相关元素与 ISO/IEC 8650-1 中定义的联系控制服务元素(ACSE)结合起来。这些元素结合的方式确保与国际标准第 5 版之前的系统编码兼容。

OSI 会话协议的相关部分在 8.3 中定义。

### 7.1 OSI-PDU

基于 OSI 的协议消息在一个 OSI 应用联系中作为目录协议数据单元被传递, 此协议数据单元由下述的 OSI-PDU 数据类型表示:

OSI-PDU {APPLICATION-CONTEXT: protocol} ::= TYPE-IDENTIFIER. &Type (  
 OsiBind {{protocol}} |  
 OsiBindResult {{protocol}} |  
 OsiBindError {{protocol}} |  
 OsiOperation {{protocol, & Operations}} |  
 PresentationAbort)

### 7.2 目录 PDU 结构

OSI 环境中的一个目录 PDU, 包含 ISO/IEC 8823-1 中定义的 OSI 表示层的协议元素, 如果相关的话, 还包含 ISO/IEC 8650-1 中定义的 ACSE 协议元素, 以及正讨论的协议的目录特定的协议元素。

OsiBind、OsiBindResult 和 OsiBindError 除了具有目录特定的协议元素外, 还具有表示协议元素和 ACSE 协议元素, 而 OsiOperation 除了具有目录特定的协议元素外, 仅具有表示协议元素。PresentationAbort 仅具有表示协议元素。

在一个特定的目录 PDU 中包含的表示层协议元素包含 PPDU。

注 1: 术语 PPDU(表示协议数据单元)在此引入, 当讨论表示协议差错时被 Abort-reason 数据类型所引用。否则, 此术语与本系列目录规范无关。

在一个特定的目录 PDU 中包含的 ACSE 协议元素组成一个 ACSE PDU。

注 2: 在 ISO/IEC 8650-1 中为 ACSE PDU 使用术语 APDU(应用协议数据单元)。但由于一个特定的目录 PDU 中的目录特定的协议元素在原则上也包含一个 APDU, 因此这里使用术语 ACSE PDU 以避免混淆。

本目录规范使用下列 PPDU:

- CP PPDU, 由 ISO/IEC 8823-1 中定义的数据类型 CP-type 来反映。它是数据类型 OsiBind 的一部分。
- CPA PPDU, 由 ISO/IEC 8823-1 中定义的数据类型 CPA-PPDU 来反映。它是数据类型 OsiBindResult 的一部分。
- CPR PPDU, 由 ISO/IEC 8823-1 中定义的数据类型 CPR-PPDU 来反映。它是数据类型 OsiBindError 的一部分。

- d) TD PPDU,由 ISO/IEC 8823-1 中定义的数据类型User-data 来反映。它是数据类型OsiOperation 的一部分。
- e) ARU PPDU,由 ISO/IEC 8823-1 中定义的ARU-PPDU 来反映。它是本目录规范中定义的数据类型ARU-PPDU 的一部分;以及
- f) ARP PPDU,由 ISO/IEC 8823-1 中定义的ARP-PPDU 来反映。它组成了本目录规范中定义的数据类型ARP-PPDU。

没有为应用联系的释放(OsiUnbind 和OsiUnbindResult)定义 PPDU。然而,ISO/IEC 8823-1 中定义的User-data 数据类型被用于携带OsiUnbind 和OsiUnbindResult。

本目录规范使用如下 ACSE PDU:

- a) AARQ-apdu 是数据类型OsiBind 的一部分;
- b) AARE-apdu 是数据类型OsiBindResult 和数据类型OsiBindError 的一部分;
- c) RLQR-apdu 是数据类型OsiUnbind 的一部分;
- d) RLRE-apdu 是数据类型OsiUnbindresult 的一部分;以及
- e) ABRT-apdu 是数据类型ARU-PPDU 的一部分。

### 7.3 会话 PDU

除了目录 PDU 外,本目录规范还定义了会话协议数据单元(SPDU)。所有的目录 PDU 都包含在一个 SPDU 中。

本目录规范使用了如下 SPDU:

- a) CONNECT SPDU 用于携带OsiBind;
- b) ACCEPT SPDU 用于携带OsiBindResult;

注:根据 ISO/IEC 8650-1:1998 的 8.1.3,AARE ACSE PDU(表示为AARE-apdu 和AAREerr-apdu)被映射为 P-CONNECT 响应/证实,并且结果被设置为‘user rejection(用户拒绝)’。根据 ISO/IEC 8823-1:1994 的 6.2.5.6,CPR PPDU 应当在表示层发出。另外,根据 ISO/IEC 8823-1:1994 的 7.1.3,CPR PPDU 在 S-CONNECT 响应和证实会话原语中携带。

- c) REFUSE SPDU 用于携带OsiBindError,并且用于根据会话层条件拒绝一个应用联系;
- d) FINISH SPDU 用于携带OsiUnbind 来发起一个应用联系的终止;
- e) DISCONNECT SPDU 用于携带OsiUnbindResult 来完成一个应用联系的终止;
- f) ABORT SPDU 除了根据会话层问题而夭折时可以独立使用外,还用于携带 ARU-PPDU 和 ARP-PPDU;
- g) ABORT ACCEPT SPDU 不携带上层信息,但它指示了对端系统已经接收到一个夭折;以及
- h) DATA TRANSFER SPDU 用于携带OsiOperation。

关于 SPDU 的详细信息在 8.3 中给出。

### 7.4 OSI 编址

OSI 为网络层及以上各层,包括表示层,都定义了地址。网络层上的地址被称为网络服务访问点(NSAP)地址。NSAP 地址的结构在 GB/T 15126—2008 中定义。运输层上的运输地址被定义为 NSAP 地址加上一个可选的运输选择因子。会话层上的会话地址被定义为运输地址加上一个可选的会话选择因子。一个表示地址被定义为会话地址加上一个可选的表示选择因子。本目录规范中仅涉及到会话选择因子和表示选择因子。

### 7.5 规程与排序

两个应用进程之间的一个应用联系通过其中一个应用进程调用 7.6.1 定义的OsiBind 而发起。始发的应用进程在此应用联系上发送任意目录 PDU 前,须等待一个OsiBindResult 来证实此应用联系已经建立。

独立于任何排序规则,始发的应用进程在调用一个OsiBind 后,可以在任意时间点发起一个ARU-

PPDU 或 ARP-PPDU (见 7.6.7)。类似的,响应的应用进程在接收到一个 OsiBind 后,可以在任意时间点发起一个 ARU-PPDU 或 ARP-PPDU。

如果接收到一个OsiBindResult,则始发的应用进程根据正在讨论的协议,可以发送包含OsiReq, OsiRes, OsiErr 和OsiRej 的 OsiOperation。

如果作为OsiBind的响应而接收到一个OsiBindError (见 7.6.3), 或者应用联系在会话层被拒绝 (见 8.3.5), 则一个应用联系将不能够被建立。

两个应用进程可能几乎同时向对方发起OsiBind。这将被认为是两个独立的应用联系建立尝试。如果两个都成功,则结果是有两个应用联系被建立。

协议差错可以在会话协议元素、表示协议元素、ACSE 协议元素,以及目录特定的协议元素中发生。

一个协议差错可以由如下情况引起:

- a) 接收到一个未识别的或非期望的 PDU;或者
- b) 在所接收的 PDU 中有一个或多个参数是无效的或非期望的。

注 1: 根据第 12 章所规定的扩展规则,未知参数应当被忽略。ISO/IEC 8823-1:1994 的 8.5 和 ISO/IEC 8650-1:1996 的 7.4 也规定了类似的规则。

注 2: ISO/IEC 8823-1:1994 的 6.4.4.2 和 6.4.4.3 区分了协议差错和无效 PPDU。由于这两种情况都引起相同类型的夭折,因此本目录规范对此不做区分。ISO/IEC 8650-1:1996 的 7.3.3.4 也没有对此进行区分。

在这两种情况下,应用联系或处于建立/终止阶段的应用联系都应夭折。

如果此问题是在会话协议中检测到的,则应当发出一个ABORT SPDU (见 8.3.8),不携带用户数据。

如果此问题是在表示协议中检测到的,则应当发出一个ARP-PPDU(见7.6.7.2)。

如果此问题是在 ACSE 协议中检测到的,则应当发出一个 ARU-PPDU,且 abort-source 被设置为 acse-service-provider (见 7.6.7.1)。

如果此问题是在目录协议中检测到的,则应当发出一个ARU-PPDU,且abort-source 被设置为acse-service-user。

## 7.6 目录 PDU 规范

### 7.6.1 OSI 绑定请求

```

OsiBind {APPLICATION-CONTEXT:Protocols)::= SET {
    mode-selector          [0] IMPLICIT SET {mode-value [0] IMPLICIT INTEGER
(1)},
    normal-mode-parameters [2] IMPLICIT SEQUENCE {
        protocol-version    [0] IMPLICIT BIT STRING {version-1(0)}
                                DEFAULT(version-1),
        calling-presentation-selector [1] IMPLICIT Presentation-selector OPTIONAL,
        called-presentation-selector [2] IMPLICIT Presentation-selector OPTIONAL,
        presentation-context-definition-list
                                [4] IMPLICIT Context-list,
        user-data            CHOICE {
            fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE
(1) OF
                                SEQUENCE {
                                    transfer-syntax-name Transfer-syntax-name OPTIONAL,
                                    presentation-context-identifier Presentation-context-identifier,

```

presentation-data-values	CHOICE {
single-ASN1-type	[0] AARQ-apdu {{Protocols}}}}}}}

Presentation-selector ::= OCTET STRING(SIZE(1..4, ..., 5..MAX))

Context-list ::= SEQUENCE SIZE (2) OF

SEQUENCE {

presentation-context-identifier	Presentation-context-identifier,
abstract-syntax-name	Abstract-syntax-name,
transfer-syntax-name-list	SEQUENCE OF Transfer-syntax-name}

Presentation-context-identifier ::= INTEGER(1..127, ..., 128..MAX)

Abstract-syntax-name ::= OBJECT IDENTIFIER

Transfer-syntax-name ::= OBJECT IDENTIFIER

AARQ-apdu { APPLICATION-CONTEXT: Protocols } ::= [APPLICATION 0] IMPLICIT SEQUENCE {

protocol-version	[0] IMPLICIT BIT STRING {version1(0)} DEFAULT {version1},
------------------	---

application-context-name	[1] Application-context-name,	
called-AP-title	[2] Name	OPTIONAL,
called-AE-qualifier	[3] RelativeDistinguishedName	OPTIONAL,
called-AP-invocation-identifier	[4] AP-invocation-identifier	OPTIONAL,
called-AE-invocation-identifier	[5] AE-invocation-identifier	OPTIONAL,
calling-AP-title	[6] Name	OPTIONAL,
calling-AE-qualifier	[7] RelativeDistinguishedName	OPTIONAL,
calling-AP-invocation-identifier	[8] AP-invocation-identifier	OPTIONAL,
calling-AE-invocation-identifier	[9] AE-invocation-identifier	OPTIONAL,
implementation-information	[29] IMPLICIT Implementation-data	OPTIONAL,
user-information	[30]	

IMPLICIT SEQUENCE SIZE(1) OF [UNIVERSAL 8] IMPLICIT SEQUENCE {

direct-reference	OBJECT IDENTIFIER OPTIONAL,
indirect-reference	Presentation-context-identifier,
encoding	CHOICE {
single-ASN1-type	[0] TheOsiBind {{Protocols}}}}}}}

注：在 ISO/IEC 8823-1:1994 中，user-information 组件被定义为一个 EXTERNAL。由于 external 的内容是已知的，因此如果提供了 EXTERNAL 的确切编码，则它可以用来辅助实现者。这里，external 的记法根据的是 GB/T 16263.1、GB/T 16263.2 和 GB/T 16263.4 中定义的编码。这不是一个完全合法的 ASN.1。使用 EXTERNAL 记法的正式、合法的 ASN.1 规范在附录 B 中提供。

Application-context-name ::= OBJECT IDENTIFIER

AP-invocation-identifier ::= INTEGER

AE-invocation-identifier ::= INTEGER

Implementation-data ::= GraphicString

TheOsiBind {APPLICATION-CONTEXT: Protocols} ::=

[16] APPLICATION-CONTEXT. &bind-operation. &ArgumentType ({Protocols})

OsiBind 被用来发起一个应用联系。OsiBind 包含表示协议元素(见 7.6.1.1), ACSE 协议元素(见 7.6.1.2)和目录绑定协议元素(见 7.6.1.3)。绑定请求的格式应当按照这些条中给出的规范进行约束。

OsiBind 在会话CONNECT SPDU(见 8.3.3)的用户数据参数或扩展的用户数据参数中携带。

应用联系的响应者应当按照如下顺序检测协议元素:

- 1) 会话协议元素应被检测。如果有一个或多个这些协议元素是不可接受的,则应当返回一个 REFUSE SPDU(见 8.3.5)。否则,继续。
- 2) 表示协议元素应被检测。如果有一个或多个这些协议元素是不可接受的,则应当返回一个 OsiBindError,且包含一个 provider-reason 组件,同时不包含 user-data 组件(见 7.6.3.1)。否则,继续。
- 3) ACSE 协议元素应被检测。如果有一个或多个这些协议元素是不可接受的,则应当返回一个 OsiBindError,且 AARErr-apdu 中的 result-source-diagnostic 组件存在,同时 user-information 组件不存在,如 7.6.3.2 中的规定。否则,继续。
- 4) 目录绑定应当根据正在讨论的目录协议的规则来进行检测。如果响应者能够接收目录绑定,则应当返回一个 OsiBindResult(见 7.6.2)。否则,应当返回一个 OsiBindError,且 AARErr-apdu 中的 user-information 组件存在。

如果在上述顺序中,任何时刻检测到一个协议差错,则应当按照 7.5 的规定发出相应的夭折。

#### 7.6.1.1 表示协议元素

组成一个 CP PPDU 的表示协议元素是由上述的 OsiBind 数据类型来定义的,除了内嵌的 AARQ-apdu 之外。

mode-selector 组件应当总是被设置为 1。

注 1: ISO/IEC 8823-1:1994 定义了表示连接的两种方式。本系列目录规范总是使用其中的 normal-mode。

normal-mode-parameters 组件具有下列子组件:

- a) protocol-version 子组件应当被忽略,或者被设置为 version-1。如果有不同的指定,则响应者应当返回一个 OsiBindError,其中 provider-reason 被设置为 protocol-version-not-supported。
- b) calling-presentation-selector 子组件,如果提供的话,其值应当从本地拥有的信息中获取。关于表示选择因子的定义,见 7.4。
- c) called-presentation-selector 子组件,如果提供的话,其值应当从下述方式中获取:
  - 作为之前目录操作结果的一个 ContinuationReference 中的 AccessPoint 值中获取到的信息(见 GB/T 16264.4—2008);或者
  - 本地拥有的信息。

如果响应者不使用表示选择因子编址,或者如果所提供的表示选择因子不是目录应用进程的,则响应者应当返回一个 OsiBindError,且 provider-reason 被设置为 called-presentation-address-unknown。

- d) presentation-context-definition-list 子组件应当具有两个元素,每个都是一个序列(sequence)类型,包括:

——一个由发起者选择的presentation-context-identifier。它应是一个奇数整数,且为两个元素所分配的应当是不同的。

——一个abstract-syntax-name:

- i) 对于其中一个元素,它应当是一个客体标识符,标识了 ACSE 抽象语法(id-acseAS);并且
- ii) 对于另一个元素,它应当是与要建立的应用联系类型相对应的目录抽象语法的客体标识符(相应的,为id-as-directoryAccessAS, id-as-directorySystemAS, id-as-directoryShadowAS 或者id-as-directoryOperationalBindingManagementAS);

——一个transfer-syntax-name-list,它应当由一个单独元素组成,此元素为基本编码规则(BER)的客体标识符;

注 2: ISO/IEC 8823-1:1994 允许建议多个传送语法,在这些传送语法中,响应者可以选择其一。在第 12 章中定义的扩展规则中要求使用 BER。

关于抽象语法和传送语法的详细信息见 8.1。

- e) user-data 子组件具有下列元素:

注 3: user-data 子组件体现了 ISO/IEC 8823-1:1994 中定义的 CP PDU 的user-data 的fully-encoded-data 选项。fully-encoded-data 包括一个PVD-list 的序列。本目录规范精确地要求仅有一个PVD-list。因此,此序列类型规定了有且仅有一个值。

——transfer-syntax-name 子组件,如果存在的话,应当是基本编码规则(BER)的客体标识符;

注 4: 根据 ISO/IEC 8823-1:1994 的 8.4.2.7:“当为表示数据值的表示上下文提议了多个传送语法的名字时,应当出现传送语法的名字”。

——presentation-context-identifier 子组件应当给定一个值,其值与规定了 ACSE 抽象语法的presentation-context-definition-list 中的元素的presentation-context-identifier 相同;

——presentation-data-values 子组件应当拥有 7.6.1.2 规定的 ACSE 协议元素。

### 7.6.1.2 ACSE 协议元素

ACSE 协议元素是上述AARQ-apdu 数据类型所定义的那些元素,除了内嵌的TheOsiBind 之外。

注 1: ACSE 协议元素是 ISO/IEC 8650-1:1996 中定义的AARQ-apdu 中的相关组件。本系列目录规范中仅使用了 ACSE 的核心功能单元。根据 ISO/IEC 8650-1:1996 的 9.1,组件sender-acse-requirements, mechanism-name, calling-authentication-value 和application-context-namelist 是不适用的。

protocol-version 组件应当被忽略,或者被设置为version1,即比特 0 被设置。如果该组件存在,则发起者不能在比特 0 后包含任何比特。如果响应者接收到一个绑定请求,其中该组件存在,且比特 0 被设置,同时有一个或多个其他比特也都被设置,则这些比特应当被忽略。如果比特 0 未被设置,但是其他某些比特被设置,则响应者应用进程应当以一个OsiBindError(见 7.6.3)作为答复,其中Associate-source-diagnostic 被设置为no-common-acse-version。

application-context-name 组件应当:

- a) 对于 DAP,被设置为id-ac-directoryAccessAC;
- b) 对于 DSP,被设置为id-as-directorySystemAC;
- c) 对于 DISP,被设置为如下之一:
  - id-ac-shadowConsumerInitiatedAC;
  - id-ac-shadowSupplierInitiatedAC;
  - id-ac-shadowSupplierInitiatedAsynchronousAC;或者
  - id-ac-shadowConsumerInitiatedAsynchronousAC;
- d) 对于 DOP,被设置为id-ac-directoryOperationalBindingManagementAC。

如果响应者不支持规定的 application-context-name, 则它应以一个 OsiBindError (见 7.6.3) 来答复, 其中 Associate-source-diagnostic 被设置为 application-context-name-not-supported。

called-AP-title 组件, 如果存在的话, 其值应当从如下方式中获取:

- 作为之前目录操作结果的 ContinuationReference 中返回的信息中; 或者
- 本地拥有的信息中。

如果响应者不能识别 called-AP-title, 则它应当以一个 OsiBindError (见 7.6.3) 来答复, 其中 Associate-source-diagnostic 被设置为 called-AP-title-not-recognized。

called-AE-qualifier 组件, 如果存在的话, 其值应当从如下方式中获取:

- 作为之前目录操作结果的 ContinuationReference 中返回的信息中; 或者
- 本地拥有的信息中。

如果响应者不能够识别 called-AE-qualifier, 则它应当以一个 OsiBindError (见 7.6.3) 来答复, 其中 Associate-source-diagnostic 被设置为 called-AE-qualifier-not-recognized。

called-AP-invocation-identifier 组件可以可选地被提供, 如果关于其值的信息从之前的应用联系中还保留的话。如果响应者不能够识别 called-AP-invocation-identifier, 则它应当以一个 OsiBindError (见 7.6.3) 来答复, 其中 Associate-source-diagnostic 被设置为 called-AP-invocation-identifier-not-recognized。

called-AE-invocation-identifier 组件可以可选地被提供, 如果关于其值的信息从之前的应用联系中还保留的话。如果响应者不能够识别 called-AE-invocation-identifier, 则它应当以一个 OsiBindError (见 7.6.3) 来答复, 其中 Associate-source-diagnostic 被设置为 called-AE-invocation-identifier-not-recognized。

calling-AP-title 组件, 如果被提供的话, 应当从本地所拥有的信息中获得。如果响应者希望确信发起者的身份, 但又不识别 calling-AP-title 时, 则它可以通过一个 OsiBindError (见 7.6.3) 来拒绝此应用联系, 其中 Associate-source-diagnostic 被设置为 calling-AP-title-not-recognized。

calling-AE-qualifier 组件, 如果被提供的话, 应当从本地所拥有的信息中获得。如果响应者希望确信发起者的身份, 但又不识别 calling-AE-qualifier 时, 则它可以通过一个 OsiBindError (见 7.6.3) 来拒绝此应用联系, 其中 Associate-source-diagnostic 被设置为 calling-AE-qualifier-not-recognized。

calling-AP-invocation-identifier 组件可以可选地被提供。一个接收系统可以忽略其值, 如果该组件存在的话。如果响应者希望确信发起者的身份, 但又不识别 calling-AP-invocation-identifier 时, 则它可以通过一个 OsiBindError (见 7.6.3) 来拒绝此应用联系, 其中 Associate-source-diagnostic 被设置为 calling-AP-invocation-identifier-not-recognized。

calling-AE-invocation-identifier 组件可以可选地被提供。一个响应系统可以忽略其值, 如果该组件存在的话。如果响应者希望确信发起者的身份, 但又不识别 calling-AE-invocation-identifier 时, 则它可以通过一个 OsiBindError (见 7.6.3) 来拒绝此应用联系, 其中 Associate-source-diagnostic 被设置为 calling-AE-invocation-identifier-not-recognized。

implementation-information 组件可以拥有与实现相关的特定信息。该信息不影响应用联系的建立规程。

user-information 组件具有下述子组件:

- a) direct-reference, 如果存在的话, 应当具有基本编码规则 (BER) 的客体标识符;
- b) indirect-reference 应当标识 7.6.1.1 的 d) 中定义的 presentation-context-definition-list 内的目录抽象语法; 以及
- c) single-ASN1-type 应当具有 7.6.1.3 中规定的绑定协议元素。

注 2: user-information 组件对应于 ISO/IEC 8650-1:1996 中定义的 AARQ-apdu 中的 user-information 组件。

该组件是一个 SEQUENCE OF EXTERNAL。本系列目录规范要求仅存在一个 EXTERNAL (见 7.6.1 中的注)。

## 7.6.1.3 绑定协议元素

TheOsiBind 应当是为讨论中的目录协议所定义的绑定请求参数。

注：绑定参数起始于 GB/T 16975.1—2000 中定义的[16]标签。

## 7.6.2 OSI 绑定结果

如果OsiBind 被接受,且响应者决定参与到此应用联系中时,响应者将会返回一个OsiBindResult。

OsiBindResult {APPLICATION-CONTEXT:Protocols} ::= SET {  
     mode-selector [0] IMPLICIT SET {mode-value [0] IMPLICIT INTEGER  
 (1)},  
     normal-modeparameters [2] IMPLICIT SEQUENCE {  
         protocol-version [0] IMPLICIT BIT STRING {version-1(0)} DEFAULT {ver-  
 sion-1},  
     responding-presentation-selector  
         [3] IMPLICIT Presentation-selector OPTIONAL,  
     presentation-context-definition-result-list  
         [5] IMPLICIT SEQUENCE SIZE (2) OF SEQUENCE {  
             result [0] IMPLICIT Result (acceptance),  
             transfer-syntax-name [1] IMPLICIT Transfer-syntax-name},  
     user-data CHOICE {  
         fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE  
 (1) OF SEQUENCE {  
             transfer-syntax-name Transfer-syntax-name OPTIONAL,  
             presentation-context-identifier Presentation-context-identifier,  
             presentation-data-values CHOICE {  
                 single-ASN1-type [0] AARE-apdu {{Protocols}}}}}}}  
 Result ::= INTEGER {  
     acceptance (0),  
     user-rejection (1),  
     provider-rejection (2)}

AARE-apdu {APPLICATION-CONTEXT: Protocols} ::= [APPLICATION 1] IMPLICIT SE-  
 QUENCE {  
     protocol-version [0]  
         IMPLICIT BIT STRING {version1 (0)} DEFAULT {version1},  
     application-context-name [1] Application-context-name,  
     result [2] Associate-result (accepted),  
     result-source-diagnostic [3] Associate-source-diagnostic,  
     responding-AP-title [4] Name OPTIONAL,  
     responding-AE-qualifier [5] RelativeDistinguishedName OPTIONAL,  
     responding-AP-invocation-identifier [6] AP-invocation-identifier OPTIONAL,  
     responding-AE-invocation-identifier [7] AE-invocation-identifier OPTIONAL,  
     implementation-information [29] IMPLICIT Implementation-data OPTIONAL,  
     user-information [30]



IMPLICIT SEQUENCE SIZE(1) OF [UNIVERSAL 8] IMPLICIT SEQUENCE {  
 direct-reference OBJECT IDENTIFIER OPTIONAL,  
 indirect-reference Presentation-context-identifier,  
 encoding CHOICE {  
   single-ASN1-type [0] TheOsiBindRes {{Protocols}}}}}

注：见 7.6.1 中的注。

Associate-result ::= INTEGER {  
   accepted (0),  
   rejected-permanent (1),  
   rejected-transient (2)} (0..2, ...)

Associate-source-diagnostic ::= CHOICE {  
   acse-service-user [1] INTEGER {  
     null (0),  
     no-reason-given (1),  
     application-context-name-not-supported (2),  
     calling-AP-title-not-recognized (3),  
     calling-AP-invocation-identifier-not-recognized (4),  
     calling-AE-qualifier-not-recognized (5),  
     calling-AE-invocation-identifier-not-recognized (6),  
     called-AP-title-not-recognized (7),  
     called-AP-invocation-identifier-not-recognized (8),  
     called-AE-qualifier-not-recognized (9),  
     called-AE-invocation-identifier-not-recognized (10)} (0..10, ...),  
   acse-service-provider [2] INTEGER {  
     null (0),  
     no-reason-given (1),  
     no-common-acse-version (2)} (0..2, ...)}

TheOsiBindRes {APPLICATION-CONTEXT, Protocols} ::=  
   [17] APPLICATION-CONTEXT. &.bind-operation. &.ResultType ({Protocols})  
 OsiBindResult 被携带在会话 ACCEPT SPDU(见 8.3.4)的用户数据参数中。

#### 7.6.2.1 表示协议元素

组成一个 CPA PPDU 的表示协议元素是由上述的 OsiBindResult 数据类型来定义的,除了内嵌的 AARE-apdu 之外。

mode-selector 组件应当总是被设置为 1。

normal-mode-parameters 组件具有下列子组件：

- protocol-version 子组件应当被忽略,或者被设置为 version-1 ;
- responding-presentation-selector 子组件,如果被提供,其值应当从本地拥有的信息中获取;
- presentation-context-definition-result-list 子组件应当具有两个元素,其顺序对应于绑定请求的 presentation-context-definition-list 中所提供的元素顺序,每个元素都为相应元素提供上下

文协商的结果,如下所述:

- result 应当存在,并且被设置为acceptance。
- transfer-syntax-name 应当存在,并且指定了基本编码规则(BER)的客体标识符。
- d) user-data 子组件具有下列元素:
  - transfer-syntax-name 子组件,如果存在的话,应当是基本编码规则(BER)的客体标识符。
  - presentation-context-identifier 子组件应当被给定一个值,其值应与规定了 ACSE 抽象语法名字的绑定请求中的presentation-context-definition-list 中元素的presentation-context-identifier 具有相同的值。
  - presentation-data-values 子组件应当拥有 7.6.2.2 规定的 ACSE 协议元素。

#### 7.6.2.2 ACSE 协议元素

protocol-version 组件应当被忽略,或者被设置为version1,即比特0被设置。如果该组件存在,则响应者不应当在比特0之后包含任何比特。

result 组件应当被响应者设置为accepted。

result-source-diagnostic 组件应当选取acse-service-user 选项,并且取值为null 或者no-reason-given。

application-context-name 组件应当存在,并且被设置为绑定请求中相应组件的值。

responding-AP-title 组件,如果被提供,则应当从本地拥有的信息中获取。

responding-AE-qualifier 组件,如果被提供,则应当从本地拥有的信息中获取。

responding-AP-invocation-identifier 组件可以可选地被提供。响应者可以忽略此组件,如果此组件存在的话。

responding-AE-invocation-identifier 组件可以可选地被提供。响应者可以忽略此组件,如果此组件存在的话。

implementation-information 组件可以拥有与实现相关的特定信息。这些信息不会影响应用联系的建立规程。

user-information 组件具有下列子组件:

- a) direct-reference,如果存在的话,应当为ASN.1 基本编码规则(BER)分配的客体标识符。
- b) indirect-reference 应当标识 7.6.1.1 的 d)中定义的presentation-context-definition-list 内的目录抽象语法。
- c) single-ASN1-type 应当拥有 7.6.2.3 中规定的绑定结果协议元素。

#### 7.6.2.3 绑定结果协议元素

TheOsiBindRes 应当是为讨论中的目录协议所定义的绑定结果类型。

注:绑定结果以 GB/T 16975.1—2000 中定义的[17]标签开始。

#### 7.6.3 OSI 绑定差错

OsiBindError {APPLICATION-CONTEXT:Protocols} ::= CHOICE {

normal-mode-parameters SEQUENCE {

protocol-version [0] IMPLICIT BIT STRING {version-1(0)} DEFAULT {version-1},  
responding-presentation-selector

[3] IMPLICIT Presentation-selector OPTIONAL,

presentation-context-definition-result-list

[5] IMPLICIT Result-list OPTIONAL,

provider-reason [10] IMPLICIT Provider-reason OPTIONAL,

user-data CHOICE {

fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF

SEQUENCE {  
     transfer-syntax-name                   Transfer-syntax-name OPTIONAL,  
     presentation-context-identifier       Presentation-context-identifier,  
     presentation-data-values              CHOICE {  
         single-ASN1-type                  [0] AARErr-apdu {{Protocols}}}} OP-  
 TIONAL}}

Result-list ::= SEQUENCE SIZE (2) OF SEQUENCE {  
     result                                 [0] IMPLICIT Result,  
     transfer-syntax-name                  [1] IMPLICIT Transfer-syntax-name OPTIONAL,  
     provider-reason                       [2] IMPLICIT INTEGER {  
         reason-not-specified              (0),  
         abstract-syntax-not-supported      (1),  
         proposed-transfer-syntaxes-not-supported   (2)} OPTIONAL}

Provider-reason ::= INTEGER {  
     reason-not-specified                  (0),  
     temporary-congestion                  (1),  
     local-limit-exceeded                  (2),  
     called-presentation-address-unknown   (3),  
     protocol-version-not-supported       (4),  
     default-context-not-supported        (5),  
     user-data-not-readable               (6),  
     no-PSAP-available                    (7)}

AARErr-apdu { APPLICATION-CONTEXT: Protocols } ::= [APPLICATION 1] IMPLICIT SEQUENCE {

    protocol-version                      [0] IMPLICIT BIT STRING {version1(0)}  
   DEFAULT {version 1},  
     application-context-name             [1] Application-context-name,  
     result                                [2] Associate-result (rejected-permanent, . rejected-transi-  
 ent),  
     result-source-diagnostic             [3] Associate-source-diagnostic,  
     responding-AP-title                  [4] Name                                 OPTIONAL,  
     responding-AE-qualifier              [5] RelativeDistinguishedName       OPTIONAL,  
     responding-AP-invocation-identifier   [6] AP-invocation-identifier           OPTIONAL,  
     responding-AE-invocation-identifier   [7] AE-invocation-identifier           OPTIONAL,  
     implementation-information           [29] IMPLICIT Implementation-data   OPTIONAL,  
     user-information                     [30]  
         IMPLICIT SEQUENCE SIZE(1) OF [UNIVERSAL 8] IMPLICIT SEQUENCE {  
             direct-reference               OBJECT IDENTIFIER OPTIONAL,  
             indirect-reference             Presentation-context-identifier,  
             encoding                       CHOICE {

single-ASN1-type [0] TheOsiBindErr {{Protocols}}} OPTIONAL}}

注：见 7.6.1 中的注。

TheOsiBindErr {APPLICATION-CONTEXT:Protocols}::=

[18] APPLICATION-CONTEXT. &bind-operation. &Errors. &ParameterType  
({{Protocols}})

OsiBindError 被携带在会话 REFUSE SPDU(见 8.3.5)的原因代码字段中。

### 7.6.3.1 表示协议元素

组成一个 CPR PPDU 的表示协议元素是由上述的 OsiBindError 数据类型来定义的,除了内嵌的 AAREerr-apdu 之外。

normal-mode-parameters 组件具有下列子组件：

注 1：CPR-PPDU 可在 X.410 方式和常规方式之间进行选择。本系列目录规范仅使用常规方式。但依然保留 CHOICE 声明,以便确保当使用非 BER 或其他类似编码时可以按位后向兼容。

- a) protocol-version 子组件应当按照 7.6.2.1 的规定。
- b) responding-presentation-selector 子组件,如果被提供,则应当按照 7.6.2.1 的规定。
- c) presentation-context-definition-result-list 子组件应当按照如下规定：
  - 如果拒绝是与表示上下文协商无关的,则 result 元素应当被设置为 acceptance, transfer-syntax-name 应当存在,并指定基本编码规则(BER)的客体标识符,同时 provider-reason 元素应当不存在；
  - 如果讨论中的抽象语法不被任何提议的传送语法所支持,则 result 元素应当被设置为 provider-rejection, 且 provider-reason 元素应当存在,并具有适当的值；或者
  - 如果讨论中的抽象语法根本不被支持,且之前的项都不适用时,则 result 元素应当被设置为 user-rejection, 且 provider-reason 元素应当存在,并具有适当的值。
- d) 如果是由于在绑定请求的表示协议元素内检测到问题而使得应用联系被拒绝时, provider-reason 子组件应当存在。否则,该组件应当不存在。

注 2：ISO/IEC 8823-1:1994 的 6.2.4.9 对 provider-reason 的声明如下：“如果该字段存在,则表明拒绝是由响应方表示服务提供者提出的；如果该字段不存在,则表明拒绝是由响应方 PS 用户提出的。”

- e) 如果 provider-reason 子组件存在,则 user-data 子组件应当不存在。否则,该子组件应当存在,且具有下列元素：
  - transfer-syntax-name 子组件,如果存在,则应当是 ASN.1 基本编码规则(BER)的客体标识符。
  - presentation-context-identifier 子组件应当被给定一个值,其值应与规定了 ACSE 抽象语法名字的绑定请求中的 presentation-context-definition-list 中元素的 presentation-context-identifier 具有相同的值。
  - presentation-data-values 子组件应当拥有 7.6.3.2 中规定的 ACSE 协议元素。

### 7.6.3.2 ACSE 协议元素

protocol-version 组件应当按照 7.6.2.2 的规定。

application-context-name 组件应当存在,并且被设置为绑定请求中相应组件的值。

result 组件应当基于本地考虑,被设置为 rejected-permanent 或者 rejected-transient。

注：根据 GB/T 16975.1—2000 中的 11.1.1,一个绑定差错被携带在 A-ASSOCIATE 响应/证实中,其中 A-ASSOCIATE 服务原语中的结果参数值被设置为“rejected (permanent)”或者“rejected (transient)”,且绑定操作中的差错值被映射到这些服务原语的用户信息参数中。在协议级别,则被翻译为 result 组件,该组件或者被设置为 rejected-permanent, 或者被设置为 rejected-transient。大多数绑定差错都反映了一种永久条件。然而,问题为 unavailable 的 serviceError 可以被认为是暂时的。

result-source-diagnostic 组件应当根据条件,具有下述取值：

a) 如果拒绝是在一个目录协议之内,则应当选取acse-service-user选项,且取值为null或者no-reason-given;或者

b) 如果拒绝是与ACSE相关的,或者是由于指定的应用进程名称,应用实体标题或应用上下文中的差错而引起的,则应当选取acse-service-user选项,且取适当的值。

responding-AP-title组件,如果存在,则其值应当从本地拥有的信息中获取。

responding-AE-qualifier组件,如果存在,则其值应当从本地拥有的信息中获取。

responding-AP-invocation-identifier组件,如果存在,则可以被忽略,或者为将来与该DSA进行连接而保留。

responding-AE-invocation-identifier组件,如果存在,则可以被忽略,或者为将来与该DSA进行连接而保留。

implementation-information组件可以拥有与实现相关的特定信息。

user-information组件具有下列子组件:

a) direct-reference,如果存在,应当拥有ASN.1基本编码规则(BER)的客体标识符;

b) indirect-reference应当标识7.6.1.1的d)中定义的presentation-context-definition-list内的目录抽象语法;

c) single-ASN1-type应当拥有7.6.3.3中规定的绑定差错协议元素。

#### 7.6.3.3 绑定差错协议元素

TheOsiBindErr应当是与差错类型相关的绑定差错类型。

注:绑定差错起始于GB/T 16975.1—2000中定义的[18]标签

#### 7.6.4 OSI解绑定请求

OsiUnbind ::= CHOICE {

fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SEQUENCE {

presentation-context-identifier Presentation-context-identifier,

presentation-data-values CHOICE {

single-ASN1-type [0] TheOsiUnbind}}}

TheOsiUnbind ::= [APPLICATION 2] IMPLICIT SEQUENCE {

reason [0] IMPLICIT Release-request-reason OPTIONAL}

Release-request-reason ::= INTEGER {

normal (0)}

OsiUnbind被携带在会话FINISH SPDU(见8.3.6)的用户数据中。

仅有应用联系的发起者才可以调用一个解绑定请求。

注1: GB/T 16975.1—2000的8.5定义了一个CONNECTION-PACKAGE信息客体类,其中的字段&-responderCanUnbind规定了响应者是否可以发起一个解绑定请求。其缺省值为FALSE。本目录规范的第4版没有为任何协议增加&-responderCanUnbind字段。IDM协议允许响应者发起一个解绑定请求,除了DAP协议之外(见9.2.2)。

注2: GB/T 16975.1—2000的8.5还在CONNECTION-PACKAGE信息客体类中定义了一个&-unbindCanFail字段,其缺省值为FALSE。本目录规范的第4版没有为任何协议增加&-unbindCanFail字段。

#### 7.6.4.1 表示协议元素

表示协议元素仅指那些由ISO/IEC 8823-1:1994定义的数据类型User-data所定义的协议元素。

presentation-context-identifier组件应当被给定一个值,其值应与规定了ACSE抽象语法的绑定请

求中的presentation-context-definition-list 中元素的presentation-context-identifier 具有相同的值。

presentation-data-values 组件应当拥有 7.6.4.2 中规定的 ACSE 协议元素。

#### 7.6.4.2 ACSE 协议元素

reason 组件应当被设置为normal 或该组件不存在。reason 组件的不存在表示正常释放。

注 1: 根据 GB/T 16975.2—1997 的 11.1.2, reason 应当总是被设置为normal。

注 2: 根据 ISO/IEC 8823-1:1994, 一个连接的正常释放是没有表示协议元素的。正常释放是通过底层会话连接的正常释放而完成的。

#### 7.6.5 OSI 解绑定结果

```
OsiUnbindResult ::= CHOICE {
    fully-encoded-data      [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SE-
    QUENCE {
        presentation-context-identifier Presentation-context-identifier,
        presentation-data-values CHOICE {
            single-ASN1-type      [0] TheOsiUnbindRes}}

```

```
TheOsiUnbindRes ::= [APPLICATION 3] IMPLICIT SEQUENCE {
    reason [0] IMPLICIT Release-response-reason OPTIONAL}

```

```
Release-response-reason ::= INTEGER {
    normal      (0)}

```

注: 第 5 版之前的规范规定了 GB/T 16688 中定义的 A-RELEASE 服务的结果参数应当被设置“affirmative”。

OsiUnbindResult 被携带在会话 DISCONNECT SPDU(见 8.3.7)的用户数据中。

##### 7.6.5.1 表示协议元素

表示协议元素仅指的是那些被 ISO/IEC 8823-1:1994 中定义的 User-data 数据类型所定义的协议元素。

presentation-context-identifier 组件应当被给定一个值,其值应与规定了 ACSE 抽象语法的绑定请求中的presentation-context-definition-list 中元素的presentation-context-identifier 具有相同的值。

presentation-data-values 组件应当拥有 ACSE 释放请求。

##### 7.6.5.2 ACSE 协议元素

原因(reason)组件的不存在表示正常释放。

#### 7.6.6 OSI 操作

```
OsiOperation {OPERATION:Operations} ::= CHOICE {
    fully-encoded-data      [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SE-
    QUENCE {
        presentation-context-identifier Presentation-context-identifier,
        presentation-data-values CHOICE {
            single-ASN1-type      [0] CHOICE {
                request OsiReq {{Operations}},
                result   OsiRes {{Operations}},
                error     OsiErr {{Operations}},
                reject    OsiRej}}}}

```

OsiOperation 被携带在会话 DATA TRANSFER SPDU(见 8.3.10)的用户信息字段中。

##### 7.6.6.1 表示协议元素

presentation-context-identifier 组件宜给定一个值,其值与绑定请求中presentation-context-defini-

tion-list 元素的presentation-context-identifier 具有相同的值,该绑定请求规定了有关问题的目录协议的目录抽象语法名(称)。

presentation-data-values 组件应保持目录请求、结果、差错或拒绝。

#### 7.6.6.2 OSI 请求

OsiReq{OPERATION:Operations} ::= [1] IMPLICIT SEQUENCE {  
     invokeld     Invokeld,  
     opcode       OPERATION. &operationCode ({Operations}),  
     argument     OPERATION. &ArgumentType ({Operations} {@opcode})}

注 1: 请求起始于 GB/T 16975.1—2000 中定义的[1]标签。

invokeld 组件标识了一个特定的调用。它的取值不应是已经用于先前请求的值,该请求要求一个响应(结果和/或差错)并仍在进行过程中。如果出现了这种情况,则接收者应当发出一个OsiReject,其中InvokeProblem 被设置为duplicateInvocation。如果该请求不是应要求有响应,则在多长时间后可以重用invokeld 是由内部进行选择的。

注 2: 所有目前定义的目录操作都要求有一个响应。

opcode 组件的取值应为特定操作类型的操作代码。如果指定了一个未知的操作代码,则接收者应当发出一个OSIReject,其中InvokeProblem 被设置为unrecognizedOperation。

argument 组件应持有变元,该变元由所涉及协议的 opcode 组件标识的操作类型中的&ArgumentType 字段形成。

#### 7.6.6.3 OSI 结果

OsiRes {OPERATION:Operations} ::= [2] IMPLICIT SEQUENCE {  
     Invokeld     Invokeld,  
     result       SEQUENCE {  
         opcode     OPERATION. &operationCode ({Operations}),  
         result     OPERATION. &ResultType ({Operations} {@opcode})}

注: 结果起始于 GB/T 16975.1—2000 中定义的[2]标签。

invokeld 组件的取值应当与相应请求中指定的值相同。

opcode 组件的取值应当与相应请求中指定的值相同。

result 组件的取值应当是根据所讨论协议的 opcode 组件指定的操作类型中的&ResultType 字段来构造的结果。

#### 7.6.6.4 OSI 差错

OsiErr {OPERATION:Operations} ::= [3] IMPLICIT SEQUENCE {  
     invokeld     Invokeld,  
     errcode       OPERATION. &Errors. &errorCode ({Operations}),  
     error          OPERATION. &Errors. &ParameterType ({Operations} {@.errcode})}

注: 差错起始于 GB/T 16975.1—2000 中定义的[3]标签。

invokeld 组件的取值应当与相应的OsiRequest 中指定的值相同。

errcode 组件的取值应当被设置为某种差错代码,该差错是由相应的OsiRequest 中的opcode 所标识的OPERATION 信息客体中的ERRORS 字段所标识的差错之一。

error 组件持有由errcode 组件所标识的参数。

#### 7.6.6.5 OSI 拒绝

类型OsiRej 被用于报告其他目录 PDU 的差错使用。它被规定如下:

OsiRej ::= [4] IMPLICIT SEQUENCE {  
     Invokeld     Invokeld,

```

problem CHOICE {
    general          [0] GeneralProblem,
    invoke           [1] InvokeProblem,
    returnResult     [2] ReturnResultProblem,
    returnError      [3] ReturnErrorProblem}}

```

注：拒绝起始于 GB/T 16975.1—2000 中定义的[4]标签。

invokeId 组件的取值应当与被拒绝的 PDU 中指定的调用号相同，除非如果 invokeId 不能够被终止，则该字段应当选取 absent 选项（见 6.2）。

problem 组件应当拥有 7.6.6.6 中定义的拒绝问题。

#### 7.6.6.6 拒绝问题

```

GeneralProblem ::= INTEGER {
    unrecognizedPDU          (0),
    mistypedPDU              (1),
    badlyStructuredPDU       (2)}

```

GeneralProblem 是目录 PDU 的具有格式或结构的基本问题。其可能性规定如下：

- unrecognizedPDU : PDU 的引导标签指示出它不是一个 OsiRequest, OsiResult, OsiError 或 OsiReject ;
- mistypedPDU : PDU 的结构与相应的定义不符合；或者
- badlyStructuredPDU : 基于所期望的抽象语法，PDU 的结构不能被确定。

```

InvokeProblem ::= INTEGER {
    duplicateInvocation        (0),
    unrecognizedOperation      (1),
    mistypedArgument           (2),
    resourceLimitation         (3),
    releaseInProgress          (4)}

```

InvokeProblem 指示 OsiRequest 中的某些组件是错误的。其可能性规定如下：

- duplicateInvocation : 见 7.6.6.2；
- unrecognizedOperation : 操作代码不在所涉及的目录协议定义的代码之内；
- mistypedArgument : 变元没有按照 opcode 组件所标识的操作的 &ArgumentType 字段所要求的那样组织；
- resourceLimitation : 由于资源的限制，预期的执行者不愿意执行此操作；或者
- releaseInProgress : 由于正准备释放应用联系，预期的执行者不愿意执行此操作。

```

ReturnResultProblem ::= INTEGER {
    unrecognizedInvocation      (0),
    resultResponseUnexpected     (1),
    mistypedResult              (2)}

```

ReturnResultProblem 指示一个 OsiResult 中的某些组件是错误的。其可能性规定如下：

- unrecognizedInvocation : InvokeId 没有标识正在执行中的某个请求；
- resultResponseUnexpected : 为某个操作接收到了一个结果，而这个操作并没有定义结果；

注 1：所有当前定义的目录操作类型都规定了一个结果。

- mistypedResult : 结果没有按照 opcode 组件标识的操作中的 &ResultType 字段所要求的那样组织。

```

ReturnErrorProblem ::= INTEGER {

```



UnrecognizedInvocation (0),  
 errorResponseUnexpected (1),  
 unrecognizedError (2),  
 unexpectedError (3),  
 mistypedParameter (4)}

ReturnErrorProblem 指示一个OsiError中的某些组件是错误的。其可能性规定如下：

- a) unrecognizedInvocation : InvokeId 没有标识正在执行中的某个请求；
- b) errorResponseUnexpected : 为某个操作接收到了一个差错，而这个操作并没有定义差错；  
 注 2：所有当前定义的目录操作类型都规定了一个或多个差错。
- c) unrecognizedError : 接收到了一个差错，但该差错不是本系列目录规范所规定的差错之一；
- d) unexpectedError : 接收到了一个差错，但该差错不是opcode 组件标识的操作中的&Errors 字段所标识的差错之一；或者
- e) mistypedParameter : 差错结果的参数没有按照 errcode 组件标识的差错中的&ParameterType 字段所要求的那样组织。

### 7.6.7 表示夭折

夭折可以由于某种应用问题(ARU-PPDU)而引起，或者由于某个表示层问题(ARP-PPDU)而引起。

PresentationAbort ::= CHOICE {  
     aru-ppdu ARU-PPDU,  
     arp-ppdu ARP-PPDU}

#### 7.6.7.1 OSI 应用夭折

ARU-PPDU ::= CHOICE {  
     normal-mode-parameters [0] IMPLICIT SEQUENCE {  
         presentation-context-identifier-list [0] IMPLICIT Presentation-context-identifier-list,  
         user-data CHOICE {  
             fully-encoded-data [APPLICATION 4] IMPLICIT SEQUENCE SIZE (1) OF SEQUENCE {  
                 presentation-context-identifier Presentation-context-identifier,  
                 presentation-data-values CHOICE (  
                     single-ASN1-type [0] ABRT-apdu}}}}}

Presentation-context-identifier-list ::= SEQUENCE SIZE (1) OF SEQUENCE {  
     presentation-context-identifier Presentation-context-identifier,  
     transfer-syntax-name Transfer-syntax-name}

ABRT-apdu ::= [APPLICATION 4] IMPLICIT SEQUENCE {  
     abort-source ABRT-source}

ABRT-source ::= INTEGER {  
     acse-service-user (0),  
     acse-service-provider (1)}

如果夭折是由于目录协议层的问题或是在 ACSE 内部引起的，而不是在表示协议元素内部引

起的,则ABRT-PPDU将被使用。

ABRT-PPDU被携带在会话ABORT SPDU的用户数据内,且“Transport Disconnect”字段的第2个比特应当被置位,而第3个比特应当被复位(见8.3.8)。

ABRT-PPDU可以引起运输中信息的丢失。

在一个支持DAP的连接上接收到一个ABRT-PPDU,则会终止所有的请求处理。但在GB/T 16264.4—2008中描述的某些条件外,这一点对于DSP也是正确的。确认对DIB所要求的修改是否已经发生,是目录用户的责任。

在一个支持DISP的连接上接收到一个ABRT-PPDU,其行为在ISO/IEC 9594-9:2005中描述。

在一个支持DOP的连接上接收到一个ABRT-PPDU,其行为在GB/T 16264.4—2008中描述。

#### 7.6.7.1.1 表示协议元素

normal-mode-parameters 组件具有下列子组件:

- a) presentation-context-identifier-list 子组件指示了为用户数据采用哪种传送语法。在用户数据中仅包含ACSE信息。它应当仅具有一个元素,该元素为序列类型,包括:
  - presentation-context-identifier 子组件应当被给定一个值,其值应与规定了ACSE抽象语法名字的绑定请求中的presentation-context-definition-list 中元素的presentation-context-identifier 具有相同的值;
  - transfer-syntax-name 取值应当是基本编码规则(BER)的客体标识符。
- b) user-data 子组件具有下列元素:
  - presentation-context-identifier 子组件当被给定一个值,其值应与规定了ACSE抽象语法的presentation-context-definition-list 中元素的presentation-context-identifier 具有相同的值;
  - presentation-data-values 子组件应当拥有7.6.7.1.2中定义的ACSE协议元素。

#### 7.6.7.1.2 ACSE 协议元素

如果夭折是在目录协议层引起的,则ABRT-source 应当被设置为acse-service-user。如果夭折是在ACSE层引起的,则ABRT-source 应当被设置为acse-service-provider。

注:ISO/IEC 8650-1:1996中定义的ABRT-apdu具有两个附加的参数。如果仅有Kernel被使用,这意味着夭折的使用仅仅是为了标识一个协议差错,则abort-diagnostics 不应当出现。由于本系列目录规范没有提供夭折信息,因此user-information 没有被使用。

#### 7.6.7.2 OSI 表示夭折

ARP-PPDU ::= SEQUENCE{

provider-reason	[0] IMPLICIT Abort-reason OPTIONAL,
event-identifier	[1] IMPLICIT Event-identifier OPTIONAL}

Abort-reason ::= INTEGER{

reason-not-specified	(0),
unrecognized-ppdu	(1),
unexpected-ppdu	(2),
unexpected-session-service-primitive	(3),
unrecognized-ppdu-parameter	(4),
unexpected-ppdu-parameter	(5),
invalid-ppdu-parameter-value	(6)}

Event-identifier ::= INTEGER {

cp-PPDU	(0),
cpa-PPDU	(1),
cpr-PPDU	(2),
aru-PPDU	(3),
arp-PPDU	(4),
td-PPDU	(7),
s-release-indication	(14),
s-release-confirm	(15)}

如果夭折是由于表示协议层内的问题而引起的,则ARP-PDU 将被使用。

ARP-PDU 被携带在会话 ABORT SPDU 的用户数据中,且“Transport Disconnect”字段的第 2 个比特应当被置位,而第 3 个比特应当被复位(见 8.3.8)。

ARP-PDU 可引起运输中信息的丢失。

接收到一个ARP-PDU 应当同 7.6.7.1 中为ARU-PDU 规定的那样处理。

provider-reason 组件可以取如下的值之一:

- a) reason-not-specified ;
- b) unrecognized-ppdu 指示接收到一个未知的 PPDU;

注:这可以是 ISO/IEC 8823-1:1994 中定义的 PPDU,但本目录规范中并没有使用。

某些实现中可以标识为一个unexpected-ppdu。然而,在实现时并不要求,应识别出本目录规范中没有定义的 PPDU。

- c) unexpected-ppdu 指示接收到一个由Event-identifier 所标识的 PPDU,而且是无序的;
- d) unexpected-session-service-primitive 同Event-identifier 指示的一样;
- e) unrecognized-ppdu-parameter :根据扩展规则(见 7.5 的注 1),不应当被使用;
- f) unexpected-ppdu-parameter 指示尽管有一个参数被识别出来了,但是该参数并不是在此特定时间或地点所期望出现在Event-identifier 所标识的 PPDU 中的;
- g) invalid-ppdu-parameter-value 指示某个参数在Event-identifier 所标识的 PPDU 中具有一个无效值。

当述涉及如上情况时,Event-identifier 字段应当出现。否则应当不出现。

- a) s-release-indication 指示应用联系被对端系统的会话层功能意外地终止了;
- b) s-release-confirm 指示应用联系被本地会话层功能意外地终止了。

## 8 目录协议映射到 OSI 服务

### 8.1 抽象语法和传送语法

作为应用联系的一部分,支撑协议的协议元素应在通信双方之间达成一致。这是通过将相关的抽象语法标识为绑定请求的一部分而完成的。每个抽象语法被分配了一个客体标识符,该标识符携带在绑定请求中。

每个目录协议都要求有两个抽象语法,一个体现的是 ACSE 协议的协议元素,而另一个体现的是实际的目录协议(即目录抽象语法)。

注:ACSE 的协议元素是第 5 版及后续版本系列目录规范的一部分。然而,为了后向兼容,还是有必要在绑定操作中标识出两种抽象语法。

目录抽象语法的客体标识符为:

id-as-directoryAccessAS	OBJECT IDENTIFIER ::=	{id-as 1}
id-as-directorySystemAS	OBJECT IDENTIFIER ::=	{id-as 2}
id-as-directoryShadowAS	OBJECT IDENTIFIER ::=	{id-as 3}

id-as-directoryOperationalBindingManagementAS OBJECT IDENTIFIER ::= {id-as 4}  
ACSE 抽象语法的标识如下:

id-acseAS OBJECT IDENTIFIER ::= {joint-iso-itu-t association-control(2) abstract-syntax(1) apdus(0) version(1)}

一个抽象语法的 ASN.1 编码规则由一个客体标识符来标识。

ASN.1 编码规则的客体标识符在 GB/T 16263.1—2006 中定义。为了方便起见,在这里提供了 BER 的客体标识符:

{joint-iso-itu-t asn1(1) basic-encoding(1)}

## 8.2 应用上下文

一个应用上下文是两个应用实体间为了支持某个应用联系而共享的一些公共规则的集合。一个应用上下文由一个应用上下文名字来标识,其格式是一个客体标识符。应用上下文名字由绑定操作来告知。

一个应用上下文使用如下 ASN.1 信息客体类来定义:

```
APPLICATION-CONTEXT ::= CLASS {
    &.bind-operation          OPERATION,
    &.Operations              OPERATION,
    &.applicationContextName OBJECTIDENTIFIER UNIQUE}
WITH SYNTAX {
    BIND-OPERATION          &.bind-operation
    OPERATIONS              &.Operations
    APPLICATION CONTEXT NAME &.applicationContextName}
```

&.bind-operation 字段用于提供告知应用上下文的绑定操作的类型。

&.Operations 字段被用来列出所有的与应用上下文相关的目录操作。

&.applicationContextName 字段被用来提供应用上下文的客体标识符。

注: 这个 ASN.1 信息客体类是 GB/T 16975.2—1997 中定义的信息客体类的一个简化版本,在这里提供是因为某些规范使用此 ASN.1 信息客体引用,而不使用所分配的客体标识符。

### 8.2.1 DAP 的应用上下文

```
directoryAccessAC APPLICATION-CONTEXT ::= {
    BIND-OPERATION          directoryBind
    OPERATIONS              {read | compare | abandon | list | search | addEntry
                             | removeEntry | modifyEntry | modifyDN}
    APPLICATION CONTEXT NAME id-ac-directoryAccessAC}
```

directoryAccessAC 应用上下文是一个定义 DAP 的应用上下文。支持该应用上下文则要求支持 id-acseAS 和 id-as-directoryAccessAS 抽象语法。

对于一个 DUA,它意味着除可能的放弃(Abandon)操作类型外,还至少应支持一种 DAP 操作类型。对于一个 DSA,它意味着应支持所有的 DAP 操作。

### 8.2.2 DSP 的应用上下文

```
directorySystemAC APPLICATION-CONTEXT ::= {
    BIND-OPERATION          dSABind
    OPERATIONS              {chainedRead | chainedCompare | chainedAbandon
                             | chainedList | chainedSearch
                             | chainedAddEntry | chainedRemoveEntry
                             | chainedModifyEntry | chainedModifyDN}
```

APPLICATION CONTEXT NAME id-ac-directorySystemAC}

directorySystemAC 应用上下文是一个定义 DSP 的应用上下文。支持该应用上下文则要求支持 id-acseAS 和 id-as-directorySystemAS 抽象语法。

它意味着支持上面所列的所有 DSP 操作。

### 8.2.3 DISP 的应用上下文

shadowSupplierInitiatedAC 应用上下文是某个应用联系的一个 DISP 应用上下文,在此应用联系上,影像更新由提供者发起,且操作方式为同步方式。

注:术语“消费者”和“提供者”用来指定两种角色。这些角色分别对应于 ISO/IEC 9594-9:2005 中使用的两个术语“影像消费者”和“影像提供者”。

```
shadowSupplierInitiatedAC APPLICATION-CONTEXT::= {
    BIND-OPERATION          dSAShadowBind
    OPERATIONS               {updateShadow
                              | coordinateShadowUpdate}
    APPLICATION CONTEXT NAME id-ac-shadowSupplierInitiatedAC}
```

```
shadowConsumerInitiatedAC APPLICATION-CONTEXT::= {
    BIND-OPERATION          dSAShadowBind
    OPERATIONS               {requestShadowUpdate
                              | updateShadow}
    APPLICATION CONTEXT NAME id-ac-shadowConsumerInitiatedAC}
```

shadowConsumerInitiatedAC 应用上下文是某个应用联系的一个 DISP 应用上下文,在此应用联系上,影像更新由消费者发起,且操作方式为同步方式。

```
shadowSupplierInitiatedAsynchronousAC APPLICATION-CONTEXT::= {
    BIND-OPERATION          dSAShadowBind
    OPERATIONS               {updateShadow
                              | coordinateShadowUpdate}
    APPLICATION CONTEXT NAME id-ac-shadowSupplierInitiatedAsynchronousAC}
```

shadowSupplierInitiatedAsynchronousAC 应用上下文是某个应用联系的一个 DISP 应用上下文,在此应用联系上,影像更新由提供者发起,且操作方式为异步方式。

### 8.2.4 DOP 的应用上下文

```
directoryOperationalBindingManagementAC APPLICATION-CONTEXT::= {
    BIND-OPERATION          dSAOperationalBindingManagementBind
    OPERATIONS               {establishOperationalBinding
                              | modifyOperationalBinding
                              | terminateOperationalBinding}
    APPLICATION CONTEXT NAME id-ac-directoryOperationalBindingManagementAC}
```

directoryOperationalBindingManagementAC 应用上下文是一个定义 DOP 的应用上下文。

## 8.3 会话层规范

### 8.3.1 会话协议数据单元(SPDU)的结构

会话协议数据单元(SPDU)包含一个 SPDU 标识符(SI)以及零个或多个参数,每个参数都通过参数标识符(PI)来标识,可能还包括一个参数值(PV)字段。相关的参数可以组合起来,因此可以通过一个参数组标识符(PGI)来标识。

一个 SPDU 的第一部分是 SPDU 标识符(SI)字段。它由一个单独的八位位组组成。其值为一个

二进制数。

长度标识符(LI)用来指示一个 SPDU 的长度、一个参数的长度或者一个参数组的长度。指示长度在 0~254 范围内的 LI 字段应当由一个八位位组组成。指示长度在 255~65535 范围内的 LI 字段应当由三个八位位组组成。

第 1 个八位位组编码为 1111 1111,而第 2 个和第 3 个八位位组应当包含相关参数字段的长度,其中高阶比特在两个八位位组中的前一个。

LI 字段的值不包括自身的长度,或者任何后续的用户信息字段的长度。

注:对于本目录规范所使用的 SPDU,仅有 DATA TRANSFER SPDU 具有一个用户信息字段。

八位位组的比特编号从 1 到 8,其中第 1 比特是最低有效位。

图 1 显示了 SPDU 不含参数的情况。ABORT ACCEPT SPDU 是一个示例。LI 字段因此取值为 0。



图 1 不带参数的 SPDU

图 2 显示了 SPDU 具有两个不同参数的情况,这两个参数每个都通过一个 PI 来标识。第一个 LI 字段指示的是 SPDU 的长度,不包括 SI 字段和 LI 字段本身。另两个 LI 字段指示的是参数的长度。

例如一个示例:如果第一个 PV 是 3 个八位位组,第二个 PV 是 4 个八位位组,则第一个 LI 字段的值为 11,第二个 LI 字段的值为 3,第三个 LI 字段的值为 4。

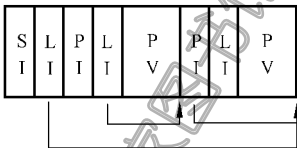


图 2 带参数的 SPDU——不成组的

图 3 显示了 SPDU 具有两个参数组合在一起的情况,每个参数都通过一个 PI 来标识。参数组通过一个 PGI 字段来标识。第一个 LI 字段指示了 SPDU 的长度,不包括 SI 字段和 LI 字段本身。下一个 LI 字段指示了参数组的长度,不包括 PGI 字段和 LI 字段本身。另两个 LI 字段指示了参数的长度。

例如一个示例:如果第一个 PV 是 5 个八位位组,第二个 PV 是 3 个八位位组,则第一个 LI 字段的值为 14,第二个 LI 字段的值为 12,第三个 LI 字段的值为 5,第四个 LI 字段的值为 3。

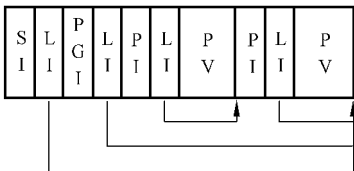


图 3 带参数的 SPDU——成组的

### 8.3.2 TSDU 尺寸和分段

运输服务数据单元(TSDU)的最大尺寸表示的是可以提交给运输层准备传输的最大八位位组数目。TSDU 的最大尺寸是在应用联系建立期间协商的,且在两个传输方向都需要进行协商(见 8.3.3 和 8.3.4)。如果一个包含了会话协议开销的目录 PDU 超出了此最大值,则有必要将目录 PDU 分段为多个 SPDU。

每个应用进程都会建议一个发起者被允许发送的最大 TSDU 的尺寸。在两个数目中,使用其中较小的一个。值为 0 被解释为对 TSDU 尺寸没有限制。如果有任意一个应用进程提议值为 0,则发起者

不能够在此应用联系上发送分段数据。

每个应用进程都会提议一个响应者被允许发送的最大 TSDU 的尺寸。在两个数目中,使用其中较小的一个。值为 0 被解释为对 TSDU 尺寸没有限制。如果有任意一个应用进程提议值为 0,则响应者不能够在此应用联系上发送分段数据。

### 8.3.3 会话 CONNECT SPDU

表 1 CONNECT SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
连接标识符	O	1	主呼 SS 用户引用	O	10	最大 64 个八位位组
			公共引用	O	11	最大 64 个八位位组
			附加的引用信息	O	12	最大 4 个八位位组
连接/接受项	M	5	协议选项	M	19	一个八位位组
			TSDU 最大尺寸	O	21	4 个八位位组
			版本号	M	22	一个八位位组
			会话用户需要	M	20	2 个八位位组
			主呼会话选择因子	O	51	最大 16 个八位位组
			被呼会话选择因子	O	52	最大 16 个八位位组
用户数据	M	193				最大 512 个八位位组
扩展的用户数据	M	194				最大 10240 个八位位组
注: M——必备的, O——可选的。						

SI 字段给定值是 13(“0D”H)。

连接标识符是一个可选的参数组,由本地产生的数据来填写,允许对此会话连接进行标识。它具有下列可选的参数:

- a) 主呼 SS 用户引用,即一个由发起者所选择的引用;

注 1: SS 用户或会话服务用户,是按照 GB/T 9387.1—1998 的表示层功能使用会话服务。

- b) 公共引用;以及

- c) 附加的引用信息。

连接/接受项是一个必选的参数组,具有下列参数:

- a) 协议选项:在本目录规范中未使用 ISO/IEC 8327-1:1996 中定义的扩展级联,所以该字段不存在,或者应当将其置为“00”H(缺省值)。然而,实现时应当接受取值为“01”H。

- b) 如果提议了一个 TSDU 最大尺寸,则 TSDU 最大尺寸的 PV 字段应当存在。如果 TSDU 最大尺寸 PV 字段存在,则:

- i) PV 字段的前两个八位位组应当包含所提议的最大 TSDU 尺寸,以八位位组来表示,其方向是从发起者到响应者,被编码为一个二进制数,其中两个八位位组中的第一个是数目的高阶部分。

- ii) PV 字段的下两个八位位组应当包含所提议的最大 TSDU 尺寸,以八位位组来表示,其方向是从响应者到发起者,被编码为一个二进制数,其中两个八位位组中的第一个是数目的高阶部分。如果该参数不存在,则对 TSDU 的最大尺寸没有限制。如果有任意一对八位位组取值为 0,则 TSDU 尺寸在这一对八位位组相关的传输方向上没有限制。

- c) 版本号——该字段应当被赋值为“02”H。

会话用户需求字段应当被设置为“0002”H。

注 2: 本目录仅使用了会话双工功能单元。

主呼会话选择因子字段取值为发起方的会话选择因子值,如果该值被分配的话,并且其值应当是根据本地拥有的信息来获取到的。如果发起方在其表示地址中没有会话选择因子,则该字段应当不存在。

如果已知被呼会话选择因子是对接收系统进行编址的一部分时,该字段应当存在。否则该字段应当不存在。如果存在,则其值宜从如下方式中获取:

- 作为之前某个目录操作结果的ContinuationReference 中返回的信息;或者
- 本地拥有的信息。

用户数据参数和扩展的用户数据参数都应当被支持,但在一个通信实例中,这两个参数中仅有一个可被使用。如果要包含的用户数据长度小于或等于 512 个八位位组,则应使用用户数据参数。如果用户数据大于 512 个八位位组,则应使用扩展的用户数据参数,而不能使用用户数据参数。

OSI 绑定请求被携带在会话 CONNECT SPDU(见 7.6.1)的用户数据中。OSI 绑定请求不应超过 10240 个八位位组。

8.3.4 会话 ACCEPT SPDU

表 2 ACCEPT SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
连接标识符	O	1	主呼用户 SS 引用	O	10	最大 64 个八位位组
			公共引用	O	11	最大 64 个八位位组
			附加的引用信息	O	12	最大 4 个八位位组
连接/接受项	O	5	协议选项	M	19	一个八位位组
			TSDU 最大尺寸	O	21	4 个八位位组
			版本号	M	22	一个八位位组
			会话用户需要	M	20	2 个八位位组
			主呼会话选择因子	O	51	最大 16 个八位位组
			响应方会话选择因子	O	52	最大 16 个八位位组
用户数据	M	193				

SI 字段给定值是 14(“0E”H)。

连接标识符是一个可选的参数组,其中填入由本地产生的、标识会话连接的数据。它可以具有下列可选的参数:

- a) 被呼 SS 用户引用;
- b) 公共引用;以及
- c) 附加的引用信息。

连接/接受项是一个必选的参数组,具有下列参数:

- a) 协议选项——该字段应当不存在,或者应当被设置为“00”H(缺省值)。然而,实现时应当接受取值为“01”H。
- b) TSDU 最大尺寸——如果响应者提议了一个 TSDU 最大尺寸,则该字段应当存在。该字段的编码和缺省值同 CONNECT SPDU(见 8.3.3)。
- c) 版本号——该字段应当被赋值为“02”H。

会话用户需求字段应当被设置为“0002”H。

如果在 CONNECT SPDU 中存在主呼会话选择因子字段,则该字段应当存在,且其值应当与相应字段的值相同。否则该字段应当不存在。

响应方会话选择因子字段,如果被提供,其值应当从本地拥有的信息中获取。



用户数据参数应当被支持。它应被用来携带OsiBindResult（见 7.6.2）。

ACCEPT SPDU 的长度不应超过 65 539 个八位位组。

### 8.3.5 会话 REFUSE SPDU

会话 REFUSE SPDU 被响应者用于拒绝一个应用联系。

表 3 REFUSE SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
连接标识符	O	1	被呼用户 SS 引用	O	9	最大 64 个八位位组
			公共引用	O	11	最大 64 个八位位组
			附加的引用信息	O	12	最大 4 个八位位组
			运输断开连接	O	17	1 个八位位组
			会话用户需求	O	20	2 个八位位组
			版本号	O	22	1 个八位位组
			原因代码	M	50	见下

SI 字段应当被赋值为 12(“0C”H)。

连接标识符是一个可选的参数组,由本地产生的数据来填写,允许对此会话连接进行标识。它可以具有下列可选的参数:

- a) 被呼 SS 用户引用;
- b) 公共引用;以及
- c) 附加的引用信息。

运输断开连接字段指示了底层的运输连接是应当被释放还是要保留。该字段的编码应当为:

- a) 比特 1 = 0:运输连接保留;
- b) 比特 1 = 1:运输连接被释放。

比特 2~8 保留。

如果该字段不存在,则运输连接被释放。

如果原因代码字段没有被设置为 2,则会话用户需求字段不应当存在。如果原因代码字段被设置为 2,则该字段应当存在,且被设置为“0002”H。

原因代码字段应当在第一个八位位组中包含一个原因代码。依赖于此第一个八位位组,可以需要使用后续的八位位组。下面的值是为第一个八位位组定义的:

- a) 0:被被呼 SS 用户拒绝;原因未指定。
- b) 1:由于临时拥塞被被呼 SS 用户拒绝。
- c) 2:被被呼 SS 用户拒绝。后续的八位位组可被用于用户数据,如果选择的是协议版本 1,则最多可为 512 个八位位组,如果选择的是协议版本 2,则最多可使得 SPDU 的总长度(包括 SI 还 LI)不超过 65539 个八位位组即可。
- d) \* 128 + 1:会话选择因子未知。
- e) \* 128 + 2:SS 用户不能附加到 SSAP。
- f) \* 128 + 3:在连接时间,会话协议机拥塞。
- g) \* 128 + 4:不支持所提议的协议版本。
- h) \* 128 + 5:被会话协议机拒绝,原因未指定。
- i) \* 128 + 6:被会话协议机拒绝;在协议实现一致性声明中声明的实现限制。

注:标识了星号(\*)的原因可被认为是永久性的,而其他可被认为是暂时的。

所有其他值都被保留。

8.3.6 会话 FINISH SPDU

表 4 FINISH SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
			运输断开连接	O	17	
用户数据	M	193				

SI 字段应当被赋值为 9。

运输断开连接字段指示了底层的运输连接是应当被释放还是要保留。该字段的编码应当为：

- a) 比特 1 = 0: 运输连接保留；或者
- b) 比特 1 = 1: 运输连接被释放。

如果该字段不存在，则运输连接被释放。

用户数据字段应当拥有 OsiUnbind (见 7.6.4)。用户数据参数的长度被限制，使得 SPDU 的全部长度(包括 SI 和 LI)不超过 65 539 个八位位组。

注：ISO/IEC 8327-1:1996 中为 FINISH SPDU 定义的附录项(Enclosure Item)参数不适用，因为仅有一个有限数目的用户数据被传递。

8.3.7 会话 DISCONNECT SPDU

表 5 DISCONNECT SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
用户数据	M	193				

SI 字段应当被赋值为 10。

用户数据字段应当拥有 OsiUnbindResult (见 7.6.5)。用户数据参数的长度被限制，使得 SPDU 的全长度(包括 SI 和 LI)不超过 65 539 个八位位组。

注：ISO/IEC 8327-1:1996 中为 DISCONNECT SPDU 定义的附录项(Enclosure Item)参数不适用，因为仅有一个有限数目的用户数据被传递。

8.3.8 会话 ABORT SPDU

表 6 ABORT SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
			运输断开连接	M	17	
			反射参数值	O	49	最大 9 个八位位组
用户数据	O	193				

SI 字段应当被赋值为 25。

运输断开连接字段指示了运输连接是否应当被保留，并跟随一个可选的原因代码。该字段的编码应当为：

- a) 比特 1 = 0: 运输连接保留；
  - b) 比特 1 = 1: 运输连接被释放；
  - c) 比特 2 = 1: 用户夭折；
  - d) 比特 3 = 1: 协议差错；
  - e) 比特 4 = 1: 无原因；
  - f) 比特 5 = 1: 在协议实现一致性声明中声明的实现限制。
- 比特 6~8 保留。

只有当运输断开连接字段指示为协议差错时,反射参数值字段才应当存在,并且应包含一个实现时定义的值和语义。

只有当运输断开连接字段指示为用户夭折时,用户数据字段才应当存在,并且应包含 ARU-PPDU (见 7.6.7.1)或者 ARP-PPDU(见 7.6.7.2)。用户数据参数的长度被限制,使得 SPDU 的全长度(包含 SI 和 LI)不超过 65 539 个八位位组。

注: ISO/IEC 8327-1:1996 中为 ABORT SPDU 定义的附录项(Enclosure Item)参数不适用,因为仅有一个有限数目的用户数据被传递。

8.3.9 会话 ABORT ACCEPT SPDU

SI 字段应当被赋值为 26。  
该 SPDU 没有相关联的参数字段。

8.3.10 会话 DATA TRANSFER SPDU

会话数据运输 SPDU 在原则上包含两个级联的 SPDU,其中第一个被称为 GIVE TOKEN SPDU。在本目录规范中,它仅以一个 SI 字段和一个长度字段的形式出现,且 SI 字段取值为 1,长度字段取值为 0。

注: ISO/IEC 8327-1:1996 定义了基本的和扩展的级联。本目录规范没有使用扩展的级联。基本级联仅适用于 DATA TRANSFER SPDU,且 ISO/IEC 8327-1:1996 的表 7 规定了 DATA TRANSFER SPDU 应当与 GIVE TOKEN SPDU 进行级联。由于我们仅使用了全双工功能单元,因此 Token 和用户数据项都不必要。

表 7 DATA TRANSFER SPDU 的参数

PGI	M/O	代码	PI	M/O	代码	长度
			附录项	O	25	1 个八位位组
用户信息字段						

DATA TRANSFER SPDU 的 SI 字段应当被赋值为 1。  
用户信息字段拥有一个目录 PDU 的全部或部分。在 SI 字段之后的 LI 字段的值不包括用户信息字段。

附录项 PV 字段,如果存在的话,应当指示该 SPDU 是否是目录 PDU 的起始或终止。如果可以用到分段的话,则该字段应当存在。如果不使用分段,该字段不应当存在。该字段的编码应当为:

- a) 比特 1 = 1:目录 PDU 的起始;  
    比特 1 = 0:不是目录 PDU 的起始;
- b) 比特 2 = 1:目录 PDU 的终止;  
    比特 2 = 0:不是目录 PDU 的终止。

比特 3~8 被保留。

如果该字段不存在,则不允许使用分段,且该 SPDU 包含一个完整的目录 PDU。

编码示例如下:

如果不包含附录项,则级联 SPDU 的编码为:“01 00 01 00”H。

如果包含附录项,则 SPDU 包含一个完整的目录 PDU,级联 SPDU 的编码为:“01 00 01 03 19 01 03”H。

8.4 运输服务的使用

在一个应用联系能够被建立之前,一个 GB/T 12453—2008 中定义的运输连接应首先被建立。

只有运输连接的发起者才被允许发起一个应用联系。

注:此限制在 ISO/IEC 8327-1:1996 的 6.1.4 中规定。

所有的会话 SPDU 都被映射为 T-DATA 请求和 T-DATA 指示。

当应用联系被拒绝时,或者已经成功连接但后来由于夭折或有序释放而被断开连接后,所支持的运输连接或者被断开,或者被重用。

在如下情况下,运输连接可被保留重用:

- a) 建立运输连接的应用进程通过 ABORT SPDU 或 FINISH SPDU 中的参数请求保持运输连接;或者
- b) 建立运输连接的应用进程接收到一个 REFUSE SPDU 或者一个 ABORT SPDU,而且其中的参数指示了运输连接将被保留。

为了避免引起对所保留的运输连接的争夺,仅有运输连接的发起者在通过一个绑定请求建立一个新的应用联系时才可重用此运输连接。

没有使用运输加速流。

## 9 IDM 协议

本章定义了互联网直接映射协议(IDM),即将请求—响应服务元素直接映射到互联网的 TCP/IP 协议,而跨越了 OSI 模型的 ACSE,表示层,会话层和运输层。协议被明显地简化,这样设计是为了实现的简单性。它是面向连接的,并且是完全异步的。

本协议使用了一定数量的协议数据单元来传送绑定、请求、响应和差错消息。

### 9.1 IDM-PDU

互联网直接映射协议的消息作为一种协议数据单元在 TCP/IP 连接上传递,该协议数据单元被称为 IDM-PDU,并且如 9.6 规定的那样被映射到 TCP/IP 上。可选的,TCP/IP 连接可通过使用 TLS 而被保护,如 9.8 规定的那样。TLS 是在 RFC 2246 和 RFC 3546 中规定的。一个 IDM-PDU 的 ASN.1 定义如下。

```
IDM-PDU {IDM-PROTOCOL:protocol} ::= CHOICE {
    bind                [0] IdmBind({protocol}),
    bindResult          [1] IdmBindResult({protocol}),
    bindError           [2] IdmBindError({protocol}),
    request             [3] Request({protocol. &Operations}),
    result              [4] IdmResult({protocol. &Operations}),
    error               [5] Error({protocol. &Operations}),
    reject              [6] IdmReject,
    unbind              [7] Unbind,
    abort               [8] Abort,
    startTLS            [9] StartTLS,
    tLSResponse         [10] TLSResponse}
```

```
IdmBind {IDM-PROTOCOL:Protocols} ::= SEQUENCE {
    protocolID          IDM-PROTOCOL. &id ({Protocols}),
    callingAETitle      [0] GeneralName OPTIONAL,
    calledAETitle       [1] GeneralName OPTIONAL,
    argument             [2] IDM-PROTOCOL. &bind-operation. &ArgumentType
                        ({Protocols}{@protocolID})}
```

```
IdmBindResult {IDM-PROTOCOL:Protocols} ::= SEQUENCE{
    protocolID          IDM-PROTOCOL. &id ({Protocols}),
    respondingAETitle   [0] GeneralName OPTIONAL,
    result              [1] IDM-PROTOCOL. &bind-operation. &ResultType}
```

((Protocols) (@protocollID))

IdmBindError{IDM-PROTOCOL;Protocols} ::= SEQUENCE {  
 protocollID IDM-PROTOCOL. &id ((Protocols)),  
 errcode IDM-PROTOCOL. &bind-operation. & Errors. & error-  
 Code  
 ((Protocols) (@protocollID)),  
 respondingAETitle [0] GeneralNameOPTIONAL,  
 aETitleError ENUMERATED {  
 callingAETitleNotAccepted (0),  
 calledAETitleNotRecognized (1)} OPTIONAL,  
 error [1] IDM-PROTOCOL. &bind-operation. & Errors. & Parame-  
 terType  
 ((Protocols) (@protocollID,@errcode))}

Request {OPERATION;Operations} ::= SEQUENCE {  
 invokelD INTEGER,  
 opcode OPERATION. &operationCode ((Operations)),  
 argument OPERATION. &ArgumentType ((Operations) (@opcode))}

IdmResult {OPERATION;Operations} ::= SEQUENCE {  
 invokelD INTEGER,  
 opcode OPERATION. &operationCode ((Operations)),  
 result OPERATION. &ResultType ((Operations) (@opcode))}

Error {OPERATION;Operations} ::= SEQUENCE {  
 invokelD INTEGER,  
 errcode OPERATION. &Errors. & errorCode ((Operations)),  
 error OPERATION. & Errors. & ParameterType  
 ((Operations) (@errcode))}

IdmReject ::= SEQUENCE {  
 invokelD INTEGER,  
 reason ENUMERATED{  
 mistypedPDU (0),  
 duplicateInvokeIDRequest (1),  
 unsupportedOperationRequest (2),  
 unknownOperationRequest (3),  
 mistypedArgumentRequest (4),  
 resourceLimitationRequest (5),  
 unknownInvokeIDResult (6),  
 mistypedResultRequest (7),  
 unknownInvokeIDError (8),

unknownError	(9),
mistypedParameterError	(10)}}

Unbind::= NULL

Abort::= ENUMERATED{

mistypedPDU	(0),
unboundRequest	(1),
invalidPDU	(2),
resourceLimitation	(3),
connectionFailed	(4),
invalidProtocol	(5),
reasonNotSpecified	(6)}

StartTLS::= NULL

TLSResponse::= ENUMERATED {

success	(0),
operationsError	(1),
protocolError	(2),
unavailable	(3)}

一个bind PDU 是在发起者和响应者之间传递一个绑定请求。protocolID 标识了将要使用的 IDM-PROTOCOL 协议(见 9.4)。argument 是所标识的协议中 BIND-OPERATION 中的 ARGUMENT 字段的值。callingAETitle 是发送 bind PDU 的本地应用实体的名字。calledAETitle 是 bind PDU 要发送到的远端应用实体的名字。

一个bindResult PDU 是一个成功的绑定请求所返回的响应。protocolID 的取值与相应的 bind PDU 中的值相同。result 是所标识的协议中 BIND-OPERATION 中的 RESULT 字段的值。respondingAETitle 是发送 bindResult 的远端应用实体的名字。

一个bindError PDU 是一个不成功的绑定请求所返回的响应。protocolID 的取值与相应的 bind PDU 中的值相同。errcode 是所标识的协议中 BIND-OPERATION 中的 ERRORS 字段所列出的差错代码之一。error 是 errcode 所标识的 ERROR 中 PARAMETER 字段的值。respondingAETitle 是发送 bindError 的远端应用实体的名字。如果被呼系统接收到一个 bind PDU,且所提供的 callingAETitle 不被被呼系统所接受,则 aETitleError 将被设置为 callingAETitleNotAccepted。如果远端应用实体接收到一个 bind PDU,且远端应用实体已知正在绑定的应用,但不接受在 bind PDU 中发送的 calledAETitle 作为自己的名字,则 aETitleError 将被设置为 calledAETitleNotRecognized。

一个request PDU 被发送用来请求一个操作。invokeID 标识了一个特定的请求以及相应的响应,它是一个正整数,且取值与之前在此 TCP/IP 连接上所发送的任何请求都不同。opcode 的取值是所选协议的 OPERATIONS 字段中列出的操作代码之一。argument 是 opcode 所标识的 OPERATION 中的 ARGUMENT 字段的值。

注: X.500 系统中的 InvokeId 在语义上与 RFC 2251 的 4.1.1.1 中定义的 LDAP 系统中的 messageID 是等价的。

一个result PDU 是一个成功的操作请求所返回的响应。invokeID 和 opcode 的取值与相应的请求 PDU 中的值相同。result 是 opcode 所标识的 OPERATION 中的 RESULT 字段的值。

一个error PDU 是一个不成功的操作请求所返回的响应。invokeID 的取值与相应的请求 PDU 中

的值相同。

errcode 是在请求 PDU 的操作中的 ERRORS 字段所列出的差错代码之一。error 是 errcode 所标识的 ERROR 中 PARAMETER 字段的值。

如果在所接收的 request, result 或 error PDU 中检测到一个协议差错, 但调用 ID 能够被恢复, 则返回一个 reject PDU 作为响应。invokeID 是所接收到的出错 PDU 的调用 ID。reason 是一个代表差错的整数代码, 在 9.4 中描述。

一个 unbind PDU 被发送是以一种有序的方式关闭一个绑定, 正如在 9.2 中描述的那样。它没有参数。

一个 startTLS PDU 被 TCP/IP 的发起者发送, 用来请求 TLS 的建立。

一个 tLSResponse PDU 是 TCP/IP 的响应者在接收到一个 startTLS PDU 后所发送的。一个成功的 tLSResponse 指示响应者愿意也能够协商 TLS。一个非成功的 tLSResponse 指示响应者或者不愿意, 或者不能够协商 TLS。当响应者检测到不正确的操作顺序时, 如在 TLS 已经被建立后又接收到一个 startTLS PDU, 在这种情况下, 响应者应当返回一个 operationsError。如果由于设计原因或目前的配置原因等, 响应者不支持 TLS, 则响应者应当返回一个 protocolError。如果响应者支持 TLS, 但它在 startTLS 请求的时间点不能够建立 TLS, 则响应者应当返回一个 unavailable。

## 9.2 顺序需求

### 9.2.1 绑定

TCP/IP 连接的发起者应当向响应者发出 bind PDU。响应者通过发送一个 bindResponse 或一个 bindErrorPDU 来作为答复。一旦接收到 bindResponse PDU, 则可以说发起者和响应者之间的一个联系就准备好了。

在发送 request PDU 之前, 发起者应当首先发送一个 bind PDU。它在发出 bind PDU 之后, 但在接收到 bindResponse 或 bindError 之前, 就可发出 request PDU。响应者应当在处理并响应所接收到的 requestPDU 之前, 首先处理并响应一个接收到的 bind PDU。

如果协议允许响应者发起一个请求, 则响应者可以在它一发送完某个 bindResponse PDU 后, 就发出这样一个请求。此时发起者应当在对所接收到的 request PDU 进行答复之前, 首先处理 bindResponse。

如果收到一个 bindError, 则发起者可选择或者通过发送一个新的绑定 PDU 请求来尝试另一个绑定, 或者选择关闭 TCP/IP 连接。

如果两个应用实体都使用绑定 PDU 中的 AETitle 信息, 则作为对绑定 PDU 的响应, 可收到一个 bindError PDU, 其中 aETitleError 被设置为 callingAETitleNotAccepted 或 calledAETitleNotRecognized。

### 9.2.2 解绑定

当使用 DAP 协议时, 仅有绑定的发起者才可以发出一个 unbind PDU。对于其他任何协议, 发起者或响应者都可发出一个 unbind PDU。一个 unbind 是破坏性的, 即任何未完成操作的结果都将被丢失 (未定义)。为了避免数据的丢失, 发起者应当在所有的请求都有响应后才能发出解绑定请求。

发起者或响应者在任何时间都可关闭底层的 TCP/IP 连接。任何未完成的请求都将丢失。

### 9.2.3 请求和响应

在发出一个 bind PDU 或一个 bindResult PDU 后, 可在任何时间发出一个 request PDU, 请求该 PDU 的接收者执行所指示的操作。request PDU 的接收者应当以一个 result, error 或者 reject PDU 来进行答复。

请求是异步的, 并不能保证响应的顺序与发出请求的顺序相同。

响应的接收者应当使用调用 ID 作为主要指示符, 来判断响应属于哪个请求, 如果调用 ID 是差错的, 则应当拒绝此响应。

#### 9.2.4 拒绝

应当使用reject PDU 来指示在处理一个request ,result 或error PDU 时,遇到了某种问题。

如果发生了任何其他协议差错,或者如果调用 ID 不能被识别,则连接将被关闭。

#### 9.3 协议

在 IDM 协议内所使用的协议通过使用IDM-PROTOCOL 信息客体类来定义,定义如下:

```
IDM-PROTOCOL::= CLASS {
    &.bind-operation      OPERATION,
    &. Operations         OPERATION,
    &.id                  OBJECT IDENTIFIER UNIQUE}

WITH SYNTAX {
    BIND-OPERATION       &.bind-operation
    OPERATIONS           &. Operations
    ID                   &.id}
```

IDM-PROTOCOL 类的每个实例都定义了 IDM 协议内使用的绑定操作和请求/响应操作。bind-Operation 字段定义了绑定所使用的操作;该操作的ARGUMENT 字段与标识该协议的bind PDU 一起使用,RESULT 字段与bindResult PDU 一起使用,并且该操作的ERRORS 字段中给定的差错之一与bindError PDU 一起使用。

Operations 字段定义了 IDM 协议的request ,result 和error PDU 内可以使用的操作。id 字段是协议标识符。

它还隐含地为一个绑定请求决定了应用上下文。因而,为每个必需的应用上下文都定义了一个单独>IDM-PROTOCOL 。

#### 9.4 拒绝原因

一个reject PDU 是为了响应各种差错条件而返回的。它们所表示的差错条件和原因代码描述如下:

当 PDU 被无效地构造时,返回一个mistypedPDU 原因。

当接收到一个请求 PDU,但在连接建立后,此invokeID 在之前已经被使用过了,则返回一个duplicateInvokeIDRequest 原因。

当接收到一个请求 PDU,但所请求的操作不支持时,则返回一个unsupportedOperationRequest 原因。

当接收到一个请求 PDU,但所请求的操作未知时,则返回一个unknownOperationRequest 原因。

当接收到一个请求 PDU,但argument 被无效的构造时,则返回一个mistypedArgumentRequest 原因。

当接收到一个请求 PDU,但由于资源限制不能执行操作时,则返回一个resourceLimitationRequest 原因。

当接收到一个结果 PDU,但其invokeID 与期望接收响应的操作的invokeID 不匹配时,则返回一个unknownInvokeIDResult 原因。

当接收到一个结果 PDU,但result 被无效的构造时,或者opcode 与相应的请求 PDU 的opcode 不匹配时,则返回一个mistypedResultRequest 原因。

当接收到一个差错 PDU,但其invokeID 与期望接收响应的操作的invokeID 不匹配时,则返回一个unknownInvokeIDError 原因。

当接收到一个差错 PDU,但所指示的差错不属于指定的协议,或者不允许作为操作的响应时,则返回一个unknownError 原因。

当接收到一个差错 PDU,但parameter 被无效的构造时,或者opcode 与相应的请求 PDU 的opcode



不匹配时,则返回一个mistypedParameterError 原因。

9.5 夭折原因

一个Abort PDU 是为了响应Reject 或BindError PDU 所不能涵盖的各种差错条件而返回的。它们所表示的差错条件和原因代码描述如下:

当所接收到的 PDU 具有一个无效结构时,则返回一个mistypedPDU 原因。

如果在连接建立之前就接收到一个请求 PDU,则返回一个unboundRequest 原因。

当一个 DSA 获得一个非IDM-PDU 的 PDU 时,则返回一个invalidPDU 原因。

当接收到一个绑定 PDU,但由于资源限制不能执行操作,如超出了最大的连接数量,则返回一个resourceLimitation 原因。

当 DSA 不能创建一个 TCP/IP 连接以便发出一个绑定 PDU 时,则返回一个connectionFailed 原因。

当接收到一个resultBind ,一个BindResult 或一个BindError PDU ,但protocolID 未知或不支持时,则返回一个invalidProtocol 原因。

当发起者或响应者由于其他任何原因希望关闭连接时,则返回一个reasonNotSpecified 原因。

注:夭折可以由于发起者的底层服务而引起,这样会导致协议不能够经过连接,例如返回一个原因为unboundRequest 的夭折,可以由于底层服务不能访问到目标系统而引起的。

9.6 映射到 TCP/IP

每个 IDM-PDU 都不受限地使用 ASN.1 的基本编码规则进行编码。然后,编码后的二进制数据被分段,并被放置在一个或多个分段中通过 TCP/IP 连接发送出去。每个分段都具有一个头,并且携带了已编码数据的下一个片段或部分。将一个IDM-PDU 分割为片段,以及任意片段的尺寸都是由发送者来选择的,并且是无关紧要的。在另一个IDM-PDU 被发送之前,一个 IDM-PDU 的所有片段都应当被发送完成。

一个分段的格式(一个IDM-PDU 的头加上片段)如下所示:

版本 (1 个八位位组)	结尾 (1 个八位位组)	长度 (4 个八位位组)	数据 (长度指定的八位位组)
-----------------	-----------------	-----------------	-------------------

版本字段指示了 IDM-PDU 以及到 TCP/IP 映射的版本。在本目录规范中描述的版本应当取值为 1。一个连接中的所有分组都应当具有相同的版本值。

注:通信双方如何协商版本号待研究。

结尾字段指示了数据是拥有一个未结束的 IDM-PDU 片段(值为 0),还是拥有完整的值或者是结尾片段(值为 1)。

长度字段指示了数据字段的长度,以八位位组为单位。它以“网络八位位组顺序”来发送,即高位有效八位位组在低位有效八位位组前发送。长度的最小值为 1。出于性能的考虑,如果长度能够使用长度字段中的 4 个八位位组来表示,则建议整个 IDM-PDU 都包含在一个分段中;如果 IDM-PDU 的长度使用 4 个八位位组还不能表示的话,则只能使用 IDM 分段了。

数据字段拥有被传递的 IDM-PDU 的下一个片段,或者如果整个值都在一个片段中传递的话,则拥有完整的 IDM-PDU。

9.7 编址

一个 IDM 风格的通信端点通过其 IP 地址和端口号来定义,并且可以以 IETF RFC 1738 的记法来表示,如:

idm://host:port

本条为这样的一个端点定义了一个等价的 OSI 网络地址格式,以便允许 IDM 协议可以与引用了 OSI 表示地址的服务定义(如目录服务定义)一起使用。一个支持 IDM 访问的系统,其表示地址的构造

正如为 OSI 访问所构造的地址那样,只是除了 P 选择因子、S 选择因子和 T 选择因子被忽略,如果它们存在的话,且网络地址采用如下规定的格式。既支持 OSI 又支持 IDM 协议栈的系统可以只拥有一个 OSI 表示地址,其中包含 OSI 和 IDM 网络地址。

为一个 IDM 端点定义的 OSI 网络地址格式遵循在 IETF RFC 1277 中所定义 的地址格式。表述为一个八位位组串,由 29 个二进制编码的十进制数字以及一个填充数字组成,如下所述:

——AFI(前两个数字)为“54”(F. 69 格式,十进制,先导 0 是有意义的)。

——IDI(下 8 个数字)为“00728722”。

——DSP(下 20 个数字)构造如下:

a) 前两个数字构成了 DSP 前缀,为 IDM 赋值为“10”。

注 1: 值 01,02,03 和 06 都已经在 IETF RFC 1277 中分配过了。03 是为 RFC 1006 协议栈赋的值。

a) 下 12 个数字是一个 4 部分的点分十进制 IP 地址,每个部分包含 3 个数字。

b) 下 5 个数字是端口号。

注 2: 端口号在 IETF RFC 1277 中是可选的,但在 IDM 中是必选的。

c) 最后一个数字是一个最后的单独十六进制的“F”,将 DSP 填充为一个完整的八位位组。

如果一个 DSA 能够在两个不同的协议栈上通信(例如基于 TCP/IP 的 IDM,或者使用 IETF RFC 1006 的基于 TCP/IP 的 OSI),则此 DSA 在其表示地址上拥有两个网络地址。例如,如果 DSA 为 IDM 协议栈使用端口 1200,而为 OSI 协议栈使用缺省端口 102,则 DSA 的 myAccessPoint 将拥有一个表示地址,包含如下:

——IDM 的网络地址 1,具有如下编码(包含自身回路 IP 地址 127. 0. 0. 1 和端口号 1200):

“54007287221012700000000101200F”H

——基于 IETF RFC 1006 的 OSI 的网络地址 2,具有如下编码(包含自身回路 IP 地址 127. 0. 0. 1;

端口号 102 是缺省分配给 IETF RFC 1006 的,因此不显式地包含在编码中,之所以允许如此,是因为在 DSP 的前缀中,使用了 03 来表示 IETF RFC 1006,而不再是表示 IDM 的 10):

“540072872203127000000001”H

## 9.8 TLS 的使用

### 9.8.1 TLS 建立

TCP/IP 连接的发起者可在任何时候通过发出一个 StartTLS PDU 来请求建立 TLS。在此请求之后,直至接收到一个 TLSResponse PDU 之前,发起者不应当再发出任何 PDU。

### 9.8.2 TLS 关闭

支持两种格式的 TLS 关闭:文雅关闭和粗鲁关闭。

#### 9.8.2.1 文雅关闭

TCP/IP 的发起者或响应者都可通过发出一个 TLS 关闭警报而终止 TLS 连接。在发送此警报时,它应当停止发送任意的 TLS 记录协议 PDU,并且在它从另一方接收到一个 TLS 关闭警报之前,对接收到的任意 TLS 记录层 PDU 都应当忽略。只有在它已经接收到 TLS 关闭警报后,才可以继续发出或接收 IDM PDU。

如果接收到一个未经请求的 TLS 关闭警报时,则接收方可选择是否保留底层的 TCP/IP 连接完好无损。如果选择保留此连接,则它应当立即以一个 TLS 关闭警报来响应,并在此后它可发送或接收 IDM PDU。

在一个 TLS 连接被关闭之后,DSA 不应当对 TLS 连接关闭之前所接收到的任何请求进行响应。

任意一方都可以在发送或接收到一个 TLS 关闭警报后,选择结束底层 TCP/IP 连接。

#### 9.8.2.2 粗鲁关闭

TCP/IP 的发起者或响应者都可通过关闭底层的 TCP/IP 连接而粗鲁地关闭一个 TLS 连接。

## 10 目录协议映射到 IDM 协议

本章给出了将目录协议映射到 IDM 协议的定义。完整的DirectoryIDMProtocols 模块在附录 E 中给出。

为清晰起见,本章重复了这些组件。

### 10.1 DAP-IP 协议

DAP-IP 协议dap-ip (基于 TCP/IP 的目录访问协议)被用于调用DirectoryAbstractService 抽象服务中的操作。它的定义如下:

DAP-IDM-PDUs::= IDM-PDU (dap-ip)

```
dap-ipIDM-PROTOCOL ::= {
    BIND-OPERATION          directoryBind
    OPERATIONS                {read | compare | abandon | list | search
                                | addEntry | removeEntry | modifyEntry | modifyDN}
    ID                        id-idm-dap}
```

本协议的操作代码和差错代码与 6.4.1 和 6.5.1 中给定的一样。

仅有 DUA 才能够使用此协议发起连接。仅有连接的发起者才能够请求此协议中的操作。

### 10.2 DSP-IP 协议

DSP-IP 协议dsp-ip (基于 TCP/IP 的目录系统协议)被用于调用DistributedOperations 抽象服务中的操作。它的定义如下:

DSP-IDM-PDUs::= IDM-PDU (dsp-ip)

```
dsp-ip IDM-PROTOCOL ::= {
    BIND-OPERATION          directoryBind
    OPERATIONS                {chainedRead | chainedCompare | chainedAbandon
                                | chainedList | chainedSearch
                                | chainedAddEntry | chainedRemoveEntry
                                | chainedModifyEntry | chainedModifyDN}
    ID                        id-idm-dsp}
```

本协议的操作代码和差错代码与 6.4.1 和 6.5.1 中给定的一样。

DSA 可以使用此协议,且连接的发起者和响应者都可以请求此协议中的操作。

### 10.3 DISP-IP 协议

DISP-IP 协议disp-ip (基于 TCP/IP 的目录信息影像协议)被用于调用DirectoryShadowAbstractService 抽象服务中的操作。它的定义如下:

DISP-IDM-PDUs::= IDM-PDU (disp-ip)

```
disp-ip IDM-PROTOCOL ::= {
    BIND-OPERATION          directoryBind
    OPERATIONS                {requestShadowUpdate
                                | updateShadow
                                | coordinateShadowUpdate}
    ID                        id-idm-disp}
```

本协议的操作代码和差错代码与 6.4.2 和 6.5.2 中给定的一样。

DSA 可以使用此协议,且连接的发起者和接收者都可以请求此协议中的操作。

#### 10.4 DOP-IP 协议

DOP-IP 协议 dop-ip (基于 TCP/IP 的目录操作绑定协议)被用于调用 OperationalBindingManagement 抽象服务中的操作。它的定义如下:

DOP-IDM-PDUs ::= IDM-PDU (dop-ip)

```
dop-ip IDM-PROTOCOL ::= {
    BIND-OPERATION          directoryBind
    OPERATIONS                { establish OperationalBinding
                                | modifyOperationalBinding
                                | terminateOperationalBinding }
    ID                        id-idm-dop }
```

本协议的操作代码和差错代码与 6.4.3 和 6.5.3 中给定的一样。

DSA 可以使用此协议,且连接的发起者和接收者都可以请求此协议中的操作。

### 11 协议栈共存

9.7 为一个 IDM 通信端点定义了一个 OSI 网络地址格式。本章推荐了一种方法,可以在支持不同协议栈的 DSA 之间共存,所支持的协议栈如 OSI, IDM 和 LDAP。为了允许在转向推荐中包含 LDAP 访问点,本章还为一个 LDAP 通信端点规范了 OSI 网络地址格式。

#### 11.1 OSI 和 IDM 协议栈之间的共存

遵循此规范的实现应当实现第 7 章和第 8 章定义的 OSI 协议栈,或者第 9 章和第 10 章定义 IDM 协议栈,或两者均实现。

如果一个链接 DSA 需要将某个请求转发到一个目标 DSA,且两个 DSA 所支持的协议栈没有共同的,则链接 DSA 应当返回一个转向推荐。该转向推荐将通过每个链接请求的 DSA 返回。如果在这些 DSA 中间,有某个 DSA 支持目标 DSA 的协议栈,则该 DSA 可以选择将请求直接发送给转向推荐所标识的目标 DSA。

如果没有一个链接 DSA 支持目标 DSA 的协议栈,则转向推荐应当返回给 DUA。该 DUA 可以将请求直接发送给目标 DSA。

如果在一个域中,混合配置了多种 DSA 产品,而某些 DSA 仅支持一种协议栈,则建议如下:

- a) 掌握仅支持一种协议栈的 DSA 知识的那些 DSA,宜支持该协议栈;或者
- b) DUA 所绑定的 DSA 应当支持两种协议栈。

#### 11.2 存在 LDAP 时的共存

支持 OSI 高层协议栈或者 IDM 协议栈的 DSA 可能也会选择支持 LDAP。这些 DSA 之间的协同工作可以通过使用链接或转向推荐来完成。而这些 DSA 与 DUA 之间的协同工作可以通过使用 LDAP 或 DAP 来完成。

对于 DSA 来说,要想为某个仅支持 LDAP 的 DUA 提供有用的转向推荐,则它应在 OSI 表示地址中表示某个潜在目标 DSA 的 LDAP 访问点。11.3 为 LDAP 定义了一个 NSAP 格式。如果一个 DSA 获得的转向推荐中包含这种类型的 NSAP,则该 DSA 能够将其转换为一个 LDAP 转向推荐,并将其返回到已连接的 LDAP 客户端。

#### 11.3 为 LDAP 定义一个 NSAP 格式

本章为一个 LDAP 通信端点定义了一个 OSI 网络地址格式,以便允许此 NSAP 与引用了 OSI 表示地址的服务定义(如目录服务定义)共同使用。一个支持 LDAP 访问的系统,其表示地址的构造正如

为 OSI 访问所构造的地址那样,只是除了 P 选择因子、S 选择因子和 T 选择因子被忽略,如果它们存在的话,且网络地址采用如下规定的格式。同时支持 OSI, IDM 以及 LDAP 协议栈的系统可以只拥有一个单独的 OSI 表示地址,其中包含 OSI, IDM 以及 LDAP 网络地址。

为一个 LDAP 端点定义的 OSI 网络地址格式遵循在 IETF RFC 1277 中所定义 的地址格式。表述为一个八位位组串,由 29 个二进制编码的十进制数字以及一个填充数字组成,如下所述:

——AFI(前两个数字)为“54”(F. 69 格式,十进制,先导 0 是有意义的)。

——IDI(下 8 个数字)为“00728722”。

——DSP(下 20 个数字)构造如下:

a) 前两个数字构成了 DSP 前缀,为 LDAP 赋值为“11”。

注 1: 值 01, 02, 03 和 06 都已经在 IETF RFC 1277 中分配过了。03 是为 RFC 1006 协议栈赋的值。10 是为 IDM 协议栈赋的值。

b) 下 12 个数字是一个 4 部分的点分十进制 IP 地址,每个部分包含 3 个数字。

c) 下 5 个数字是端口号。

注 2: 端口号在 IETF RFC 1277 中是可选的,但在 LDAP 中是必选的。

d) 最后一个数字是一个最后的单独十六进制的“F”,将 DSP 填充为一个完整的八位位组。

如果一个 DSA 能够在三个不同的协议栈上通信(例如基于 TCP/IP 的 IDM,或者使用 IETF RFC 1006 的基于 TCP/IP 的 OSI,或者 LDAP),则此 DSA 在其表示地址上拥有三个网络地址。例如,如果 DSA 为 IDM 协议栈使用端口 1200,为 OSI 协议栈使用缺省端口 102,为 LDAP 使用端口 389,则 DSA 的 myAccessPoint 将拥有一个表示地址,包含如下:

——一个 IDM 的网络地址 1,具有如下编码(包含自身回路 IP 地址 127. 0. 0. 1 和端口号 1200):

“54007287221012700000000101200F”H

——一个基于 IETF RFC 1006 的 OSI 的网络地址 2,具有如下编码(包含自身回路 IP 地址 127. 0. 0. 1; 端口号 102 是缺省分配给 IETF RFC 1006 的,因此不显式地包含在编码中,之所以允许如此,是因为在 DSP 的前缀中,使用了 03 来表示 IETF RFC 1006,而不再是表示 IDM 的 10): “540072872203127000000001”H

——一个 LDAP 的网络地址 3,具有如下编码(包含自身回路 IP 地址 127. 0. 0. 1 和端口号 389):

“54007287221112700000000100389F”H

## 12 版本及扩展规则

本章描述了版本协商规则,以及第 7 章定义的 OSI 映射协议和第 10 章定义的 IDM 映射协议的扩展规则。

目录可以是分布式的,可以有多于两个的目录应用实体(AE)进行协作来服务于一个请求。目录 AE 的实现可以遵循不同的目录服务规范的版本,这些版本可以使用不同的版本号来表示,也可以不使用。在两个直接绑定的目录 AE 中,将对版本号进行协商,协商结果为某个最高的公共版本号。

注 1: 目前每个目录协议都有两个版本。1988 版和 1993 版是属于版本 1。在第 4 版以及后续版中增加的大多数特性在第 1 版本中也是可用的。然而,某些增强的服务和协议,如签名差错,则要求在所有包含的成员中都协商为版本 2。

一个 DUA 可发出一个请求,此请求遵循该 DUA 所实现的最新目录规范的版本。使用下面定义的扩展规则,该请求应当被转发到适当的响应此请求的 DSA,而不考虑中间所经过的 DSA。响应方 DSA 应当按照下面的定义来工作。

注 2: 一个仅仅链接请求的中间 DSA 可以选择对需要执行功能的目录 PDU 中的所选元素进行检查,如名字解析。

### 12.1 DUA 到 DSA

#### 12.1.1 版本协商

当接收到一个使用了 DAP 的连接时,如绑定,则协商后的版本将仅影响 DUA 和它相连的 DSA 之

间交换协议的点对点方面。在此连接上的后续请求或响应都不受此协商后的版本的约束。

注：目前，不同的协议版本并没有指示出 DAP 中的点对点方面。

### 12.1.2 请求和响应处理

DUA 可以使用它所支持的规范的最高版本来发起一个请求。如果此请求中的一个或多个元素是关键的，则它应当在criticalExtensions 参数中指示出扩展号。

注 1：如果某个扩展所定义的值被编码为CHOICE,ENUMERATED，或INTEGER（用作ENUMERATED）类型，且如果此类型对于一个按照本规范的较早前版本所实现的 DSA 中的正常操作是必须的，则建议该扩展被标记为关键的。

当处理一个来自 DUA 的请求时，DSA 应当遵循 12.2.2 所定义的规则。

当处理一个响应时，DUA 应当：

- a) 在一个比特串中，忽略所有未知的比特名字分配；以及
- b) 如果在一个SET 或SEQUENCE 中，数字是作为一个可选元素出现的，则忽略ENUMERATED 中的所有未知的已命名数字，或者忽略被用作枚举类型的INTEGER 类型中的所有未知数字；以及
- c) 忽略 SET 中的所有未知元素，或者SEQUENCE 结尾处的所有未知元素，或者CHOICE 中的所有未知元素，若此CHOICE 本身是SET 或SEQUENCE 中的可选元素时；

注 2：作为一种本地选项，实现时可以忽略一个目录 PDU 中的某些附加元素。尤其是，SET 或SEQUENCE 中的作为必选元素的某些未知已命名数字和未知CHOICE 能够被忽略，而不会判断此操作为无效。这些元素的标识待研究。

- d) 不认为接收到未知属性类型和属性值是一种协议违例；以及
- e) 可选地，向用户报告未知的属性类型和属性值。

### 12.1.3 差错处理的扩展规则

当处理一个已知的差错类型，但所指示的问题和参数未知时，一个 DUA 应当：

- a) 不认为接收到未知问题和参数是一种协议违例（即它不应当发出一个相应的OsiReject 或Reject，或者夭折应用联系）；且
- b) 可选地，向用户上报附加的差错信息。

当处理一个未知的差错类型时，一个 DUA 应当：

- a) 不认为接收到未知差错类型是一种协议违例（即它不应当发出一个相应的OsiReject 或Reject，或者夭折应用联系）；且
- b) 可选地，向用户上报此差错。

## 12.2 DSA 到 DSA

### 12.2.1 版本协商

当建立或接收一个使用了 DSP 的连接时，如绑定，则协商后的版本将仅影响 DSA 之间交换协议的点对点方面。在此连接上的后续请求或响应都不受此协商后的版本的约束。

注 1：目前，不同的协议版本并没有指示出 DSP 中的点对点方面。

当建立或接收一个使用了 DISP 的连接时，如绑定，则协商后的版本将影响 DSA 之间交换协议的所有方面。在此连接上的后续请求或响应都受此协商后的版本的约束。

注 2：目前 DISP 协议仅有一种版本。

当建立或接收一个使用了 DOP 的连接时，如绑定，则协商后的版本将影响 DSA 之间交换协议的所有方面。在此连接上的后续请求或响应都受此协商后的版本的约束。

注 3：目前 DOP 协议仅有一种版本。

### 12.2.2 操作处理的扩展规则

如果任何一个处理某操作的 DSA（在名字解析完成后）检测到criticalExtensions 的一个元素，其语义是未知的，则它应当返回一个unavailableCriticalExtension 指示作为一个serviceError，或者放在一个

PartialOutcomeQualifier 中。

注：如果接收到一个具有一个或多个零值的criticalExtensions 字符串，则表示对应于该值的扩展在操作中不存在，或者不是关键的。在一个criticalExtensions 字符串中出现一个零值，不应当被推断为相应的扩展在目录 PDU 中是存在还是不存在。

否则，当处理一个目录 PDU 时，一个 DSA 应当：

- a) 在一个比特串中，忽略所有未知的比特名字分配；以及
- b) 如果在一个SET 或SEQUENCE 中，数字是作为一个可选元素出现的，则忽略ENUMERATED 中的所有未知的已命名数字，或者忽略被用作枚举类型的INTEGER 类型中的所有未知数字；以及
- c) 忽略SET 中的所有未知元素，或者SEQUENCE 结尾处的所有未知元素，或者CHOICE 中的所有未知元素，若此CHOICE 本身是 SET 或SEQUENCE 中的可选元素时。

### 12.2.3 链接的扩展规则

如果 PDU 是一个请求，则 DSA 应当根据名字解析过程的判断，将此包含未知类型和值的请求转发到任意的另一个 DSA。

如果 PDU 是一个响应，则 DSA 应当如同处理已知类型和值一样来处理未知类型和值（见分布式操作的目录规范中，有关结果合并的章节），并且将其转发到发起方 DSA 或 DUA。

一个按照第 5 版或后续版实现的 DSA，如果它仅仅是作为一个中间 DSA 来链接一个请求，则它应当转发具有未知操作的请求。一个按照第 5 版前的版本实现的 DSA 可以可替代地转发一个包含未知操作的请求。

### 12.2.4 差错处理的扩展规则

当处理一个已知的差错类型，但所指示的问题和参数未知时，一个 DSA 应当：

- a) 不认为接收到未知问题和参数是一种协议违例（即它不应当发出一个相应的OsiReject 或Reject，或者夭折应用联系）；且
- b) 可以尝试恢复到它所理解的差错类型，或者可以仅仅是向下一个适当的 DSA 或 DUA 返回此差错（以及未知的指示问题和参数）。

当处理一个未知差错类型时，一个仅包含在链接请求中的 DSA 应当：

- a) 不认为此未知差错类型是一种协议违例（即它不应当发出一个相应的OsiReject 或Reject，或者夭折应用联系）；以及
- b) 不尝试改正或恢复此差错以及所指示的问题和参数；以及
- c) 向下一个适当的 DSA 或 DUA 返回此未知差错类型。

当处理一个未知差错时，正在关联多个响应的 DSA 应当：

- a) 不认为此未知差错类型是一种协议违例（即它不应当发出一个相应的OsiReject 或Reject，或者夭折应用联系）；以及
- b) 不尝试改正或恢复此差错以及所指示的问题和参数；以及
- c) 将未知差错放在PartialOutcomeQualifier 中；以及
- d) 继续照常将结果关联起来。

## 12.3 客体类的扩展规则

可选的用户属性可以被加入到一个已存在的客体类中，但不分配一个新的客体标识符。

一个不支持客体类扩展的 DSA 可以拒绝这种操作，即在操作中尝试创建或修改一个条目，使得在条目中出现一个扩展的属性。

## 12.4 用户属性类型的扩展规则

用户属性类型的定义可以以这样一种不改变其匹配特性的方式进行扩展。这可以包括：

——向 ENUMERATED 类型和作为枚举类型使用的 INTEGER 类型中增加值；

——向一个比特串中增加比特。

不要求 DSA 处理包含此扩展的属性值。

DUA 不应当认为接收到这样一个扩展的属性值是一种差错。

### 13 一致性

本章定义了对本目录规范的一致性要求。

#### 13.1 DUA 的一致性

一个声明遵循本目录规范的 DUA 实现,应当满足 13.1.1 到 13.1.3 规定的要求。

##### 13.1.1 声明要求

下面内容应当被声明:

- 对 DUA 有能力调用的 directoryAccessAC 应用上下文和/或 dap-ip 协议的操作进行一致性声明;
- 对绑定安全级别(无安全、简单安全和强安全——如果是简单安全,则需要说明是否无口令,或是有口令,或是具有被保护的口令)进行一致性声明;并说明 DUA 是否能够产生签名参数或是否能够对已签名结果进行检测;
- 对在 GB/T 16264.3—2008 的表 1 中列出的,DUA 有能力发起的扩展进行一致性声明;
- 是否对基于规则的访问控制进行了一致性声明;以及
- 如果已经对强鉴权或签名操作声明了一致性,则应对证书和 CRL 扩展的标识进行一致性声明。

##### 13.1.2 静态要求

一个 DUA 应当:

- 具备能力支持第 7 章的抽象语法所定义的 directoryAccessAC 应用上下文;和/或第 10 章定义的 dap-ip 协议;
- 遵循 13.1.1 的 c) 中声明了一致性的扩展;
- 如果对基于规则的访问控制声明了一致性,则应当具备能力支持 GB/T 16264.2—2008 的 19.4 中标识的安全标签;以及
- 遵循 ISO/IEC 9594-8:2005 的第 8 章和第 15 章中的证书和 CRL 扩展,其一致性在 13.1.1 的 e) 中进行了声明。

##### 13.1.3 动态要求

一个 DUA 应当:

- 遵循第 8 章或第 10 章中定义的对已用服务的映射,或者同时遵循;以及
- 遵循 12.1 定义的扩展规程的规则。

#### 13.2 DSA 的一致性

一个声明遵循本目录规范的 DSA 实现,应当满足 13.2.1 到 13.2.3 规定的要求。

##### 13.2.1 声明要求

下面内容应当被声明:

- 对如下应用上下文和 IDM 协议进行一致性声明:directoryAccessAC, directorySystemAC, directoryOperationalBindingManagementAC, dap-ip, dsp-ip, dop-ip 或它们的组合。一个在支持分等级的操作绑定时声明了遵循 directoryOperationalBindingManagementAC 或 dop-ip 的 DSA,应当也支持 directorySystemAC 或 dsp-ip。如果一个 DSA 的知识被分散,使得到此 DSA 的知识引用被分布到多个 DSA 中,而这些 DSA 不属于此 DSA 的 DMD 时,则此 DSA 应当声明遵循 directorySystemAC 或 dsp-ip。

注 1:除了这里所说明的情况外,一个应用上下文不应当被分割;尤其是不能对特定的操作进行一致性声明。



- b) 对如下操作绑定类型进行一致性声明: shadowOperationalBindingID, specificHierarchicalBindingID, non-specificHierarchicalBindingID 或它们的组合。一个声明了遵循 shadowOperationalBindingID 的 DSA, 应当也支持在 13.3 和 13.4 指示的影像提供者和/或影像消费者的一个或多个应用上下文。
  - c) 一个 DSA 是否有能力承担在 GB/T 16264.4—2008 中定义的第一级 DSA 的角色。
  - d) 如果对 directorySystemAC 所指定的、并/或与 dap-ip 协议相关的应用上下文声明了一致性, 则需要说明是否支持 GB/T 16264.4—2008 中定义的操作链接模式。
  - e) 如果对 directoryAccessAC 所指定的、并/或与 dap-ip 协议相关的应用上下文声明了一致性, 则需要对绑定安全级别进行一致性声明(无安全、简单安全和强安全——如果是简单安全, 还需要说明是否无口令, 或是有口令, 或是具有被保护的口令); 需要说明 DSA 是否能够执行 GB/T 16264.4—2008 的 22.1 中定义的发起者鉴权; 如果可以, 需要说明是基于身份的, 还是基于签名的; 并且还需要说明 DSA 是否能够执行 GB/T 16264.4—2008 的 22.2 中定义的结果鉴权。
  - f) 如果对 directorySystemAC 所指定的、并/或与 dsp-ip 协议相关的应用上下文声明了一致性, 则需要对绑定安全级别进行一致性声明(无安全、简单安全和强安全——如果是简单安全, 还需要说明是否无口令, 或是有口令, 或是具有被保护的口令); 需要说明 DSA 是否能够执行 GB/T 16264.4—2008 的 22.1 中定义的发起者鉴权; 如果可以, 需要说明是基于身份的, 还是基于签名的; 并且还需要说明 DSA 是否能够执行 GB/T 16264.4—2008 的 22.2 中定义的结果鉴权。
  - g) 对 GB/T 16264.6—2008 中定义的选定的属性类型, 以及其他属性类型进行一致性声明, 同时需要说明对于基于语法 DirectoryString 的属性, 是否对 UniversalString, BMPString 或 UTF8String 选项进行了一致性声明。
  - h) 对 GB/T 16264.7—2008 中定义的选定的客体类, 以及其他任何客体类型进行一致性声明。
  - i) 对在 GB/T 16264.3—2008 的表 1 中所列的, DSA 有能力响应的扩展进行一致性声明。
  - j) 是否对 GB/T 16264.2—2008 的 8.9, 以及 GB/T 16264.3—2008 的 7.6, 7.8.2 和 9.2.2 中定义的集合属性进行了一致性声明。
  - k) 是否对 GB/T 16264.3—2008 的 7.6, 7.8.2 和 9.2.2 中定义的分等级属性进行了一致性声明。
  - l) 对 GB/T 16264.2—2008 中定义的操作属性类型, 以及其他任何操作属性类型进行一致性声明。
  - m) 是否对 GB/T 16264.3—2008 的 7.7.1 所描述的别名返回进行了一致性声明。
  - n) 是否对 GB/T 16264.3—2008 的 7.7.1 所描述的指示返回的条目信息是否完整进行了一致性声明。
  - o) 是否对 GB/T 16264.3—2008 的第 11.3.2 所描述的修改客体类属性, 以便增加和/或删除标识辅助客体类的值, 进行了一致性声明。
  - p) 是否对基本访问控制进行了一致性声明。
  - q) 是否对简单访问控制进行了一致性声明。
  - r) DSA 是否有能力为其 DIT 的一部分管理子模式, 正如在 GB/T 16264.2—2008 中定义的那样。
- 注 2: 管理一个子模式的能力不应当被分割; 尤其是管理特殊子模式定义的能力不应当被声明。
- s) 对 GB/T 16264.7—2008 中定义的选择名字绑定, 以及其他任何名字绑定进行一致性声明。
  - t) DSA 是否有能力管理集合属性, 正如在 GB/T 16264.2—2008 中定义的那样。
  - u) 对 GB/T 16264.6—2008 中定义的选择上下文类型, 以及其他任何上下文类型进行一致性

声明。

- v) 是否对 GB/T 16264.2—2008 的 8.8, 8.9 和 12.8, 以及 GB/T 16264.3—2008 的 7.3 和 7.6 中定义的上下文进行了一致性声明。
- w) 是否对 GB/T 16264.2—2008 的 8.5 和 9.3, GB/T 16264.3—2008 的 7.7, 以及 GB/T 16264.4—2008 定义的 RDN 中上下文的使用进行了一致性声明。
- x) 是否对 GB/T 16264.3—2008 的 7.13 定义的 DSA 信息树的管理进行了一致性声明。
- y) 是否对 ISO/IEC 9594-10:2005 定义的目录主管部门系统管理的使用进行了一致性声明。
- z) 对 ISO/IEC 9594-10:2005 中定义的选择被管客体类和管理属性类型, 以及其他任何被管客体类和属性进行一致性声明。
- aa) 是否对基于规则的访问控制进行了一致性声明。  
注 3: 对安全标签的支持要求对下列上下文的最小支持: GB/T 16264.2—2008 的 8.8 中列出的上下文, 以及 GB/T 16264.3—2008 的 7.6 中的 returnContexts。
- bb) 是否对目录操作的完整性进行了一致性声明。
- cc) 是否对 DSA 能够拥有加密和数字签名后的信息, 并可对其提供访问进行了一致性声明。
- dd) 如果对强鉴别, 签名操作, 或者被保护操作声明了一致性, 则对证书和 CRL 扩展的识别也应进行一致性声明。

### 13.2.2 静态要求

一个 DSA 应当:

- a) 具备能力支持在第 7 章定义了抽象语法的应用上下文, 以及第 10 章定义的 IDM 协议, 这些内容的一致性已经被声明;
- b) 具备能力支持在 GB/T 16264.2—2008 中的抽象语法所定义的信息框架;
- c) 遵循 GB/T 16264.4—2008 中定义的最小知识需求;
- d) 如果在一致性声明中, 可作为第一级的 DSA, 则应遵循 GB/T 16264.4—2008 中定义的支持跟上下文的需求;
- e) 具备能力支持声明了一致性的属性类型, 并且按照其抽象语法的定义;
- f) 具备能力支持声明了一致性的客体类, 并且按照其抽象语法的定义;
- g) 遵循在 13.2.1 的 i) 中声明了一致性的扩展;
- h) 如果对 GB/T 16264.2—2008 中定义的管理子模式的能力声明了一致性, 则 DSA 应当能够行使这种管理;
- i) 如果对集合属性声明了一致性, 则应具备能力执行 GB/T 16264.3—2008 的 7.6, 7.8.2 和 9.2.2 中定义的相关规程;
- j) 如果对分等级属性声明了一致性, 则应具备能力执行 GB/T 16264.3—2008 的 7.6, 7.8.2 和 9.2.2 中定义的相关规程;
- k) 具备能力支持已经声明了一致性的操作属性类型;
- l) 如果对基本访问控制声明了一致性, 则应具备能力拥有遵循基本访问控制定义的 ACI 项;
- m) 如果对简化的访问控制声明了一致性, 则应具备能力拥有遵循简化的访问控制定义的 ACI 项;
- n) 具备能力支持已经声明了一致性的上下文类型, 并且按照它们抽象语法的定义;
- o) 如果对上下文声明了一致性, 则应具备能力执行 GB/T 16264.3—2008 中定义的相关规程;
- p) 如果对在 RDN 中使用上下文声明了一致性, 则应具备能力执行 GB/T 16264.2—2008 的 9.3, GB/T 16264.3—2008 的 7.7, 以及 GB/T 16264.4—2008 中定义的相关规程;
- q) 如果对 DSA 信息树的管理声明了一致性, 则应具备能力执行 GB/T 16264.3—2008 的 7.5 和 7.13 中定义的相关规程;

- r) 如果对条目特性族的支持声明了一致性,则应具备 GB/T 16264.3—2008 的 7.3.2,7.6.4 和 7.8.3 中定义的能力;
- s) 如果对搜索放宽特性声明了一致性,则应具备 GB/T 16264.2—2008 的 13.6.2 和 GB/T 16264.3—2008 的 10.2.2 中定义的能力。特别地,一个实现应当指定:
  - 它是否支持在一个搜索请求中包含RelaxationPolicy 结构;
  - 它是否支持基于映射的匹配,匹配规则替代,或是两种均支持;以及
  - 如果它支持基于映射的匹配,则支持什么映射;
- t) 如果对分等级分组特性声明了一致性,则应具备 GB/T 16264.3—2008 的 7.5 中定义的能力;此外,实现还应声明:
  - 支持哪些分等级选项;
- u) 如果对服务的基本管理声明了一致性,则应具备 GB/T 16264.2—2008 的第 16 章中定义的能力,以及在 GB/T 16264.3—2008 的第 13 章中定义的基本检测规程。这种支持包括:
  - 支持条目计数;
  - 支持服务控制选项entryCount 和performExactly ;
  - 支持 GB/T 16264.3—2008 的 7.4 中定义的notification 扩展;
 此外,实现还应声明它是否支持:
  - 不同于自治管理点的服务特定管理点;
  - 搜索规则内的上下文特性;
  - 搜索规则内的条目工具族,同时也要求遵循该特性;
  - 上述 s)中详述的搜索规则内的搜索扩展特性,同时也要求实现时要对搜索扩展特性进行一致性声明;
  - 搜索规则内的分等级分组;
- v) 如果对目录主管部门系统管理的使用声明了一致性,则应具备能力为声明了一致性的被管客体执行 ISO/IEC 9594-10:2005 中定义的相关规程;
- w) 如果对基于规则的访问控制声明了一致性,则应具备能力拥有遵循基于规则的访问控制定义的 ACI 项;
- x) 如果对目录操作的完整性声明了一致性,则有能力为所有支持的目录操作进行签名;
- y) 如果对所存储的目录信息的完整性声明了一致性,则有能力支持attributeValueIntegrityInfo-Context 来保护目录信息;
- z) 遵循 ISO/IEC 9594-8:2005 的第 8 章中的证书和 CRL 扩展,其一致性在 13.2.1 的 dd)中进行了声明。

### 13.2.3 动态要求

一个 DSA 应当:

- a) 如果对 8.2.2,8.2.3 和 8.2.4 中定义的任何应用上下文声明了一致性,则应遵循第 8 章中定义的到已用 OSI 服务的映射;
- b) 遵循与转向推荐相关的目录分布式操作规程,正如有在 GB/T 16264.4—2008 中定义的那样;
- c) 如果对directoryAccessAC 所规定的,和/或与dap-ip 协议相关的应用上下文声明了一致性,则应遵循 GB/T 16264.4—2008 中的与 DAP 的转向推荐模式相关的规程;
- d) 如果对directorySystemAC 所规定的,和/或与 dsp-ip 协议相关的应用上下文声明了一致性,则应遵循交互的转向推荐模式,正如有在 GB/T 16264.4—2008 中定义的那样;
- e) 如果对交互的链接模式声明了一致性,则应遵循交互的链接模式,正如有在 GB/T 16264.4—2008 中定义的那样;

注:仅在这种情况下,对于一个 DSA 而言,能够调用directorySystemAC 和/或dsp-ip 的操作才是必须的。

- f) 遵循 12.2 定义的扩展规程的规则；
- g) 如果对基本访问控制声明了一致性，则应具备能力按照基本访问控制的规程来保护 DSA 内的信息；
- h) 如果对简化的访问控制声明了一致性，则应具备能力按照简化的访问控制的规程来保护 DSA 内的信息；
- i) 如果对 shadowOperationalBindingID 声明了一致性，则应遵循 ISO/IEC 9594-9:2005 和 GB/T 16264.2—2008 中与 DOP 相关的规程；
- j) 如果对 specificHierarchicalBindingID 声明了一致性，则应遵循 GB/T 16264.4—2008 和 GB/T 16264.3—2008 中与特定的分等级操作绑定相关的规程；
- k) 如果对 non-specificHierarchicalBindingID 声明了一致性，则应遵循 GB/T 16264.4—2008 和 GB/T 16264.2—2008 中与非特定分等级操作绑定相关的规程；
- l) 如果对在 RDN 中使用上下文声明了一致性，则应遵循 GB/T 16264.2—2008 的 9.4，GB/T 16264.4—2008 的 10.3, 10.4, 10.6, 10.10, 10.11 和 15.5.4 中定义的包含上下文的名字解析；
- m) 如果对基于规则的访问控制声明了一致性，则应具备能力按照基于规则的访问控制的规程来保护 DSA 内的信息；
- n) 如果对服务的基本管理声明了一致性，则应具备能力处理 GB/T 16264.4—2008 的 19.3.2 中规定的搜索规则。

### 13.3 影像提供者的一致性

一个声明遵循本目录规范的，并承担影像提供者角色的 DSA 实现，应当满足 13.3.1 到 13.3.3 规定的要求。

#### 13.3.1 声明要求

下面内容应当被声明：

- a) 对下列作为影像提供者的应用上下文进行一致性声明：shadowSupplierInitiatedAC，shadowConsumerInitiatedAC，shadowSupplierInitiatedAsynchronousAC，shadowConsumerInitiatedAsynchronousAC 和 disp-ip。一个在一致性声明中声明为影像提供者，但不支持 disp-ip 的 DSA 实现，应当最少或者支持 shadowSupplierInitiatedAC，或者支持 shadowConsumerInitiatedAC。如果该 DSA 支持 shadowSupplierInitiatedAC，则它可以可选地支持 shadowSupplierInitiatedAsynchronousAC。如果该 DSA 支持 shadowConsumerInitiatedAC，则它可以可选地支持 shadowConsumerInitiatedAsynchronousAC。如果声明了遵循 disp-ip，则它应当被声明其实现是否有能力调用 requestShadowUpdate 操作，或是否有能力响应 coordinateShadowUpdate，或者两者均可。
- b) 对安全级别进行一致性声明（无安全、简单安全和强安全）。
- c) 支持 UnitOfReplication 的级别。尤其是，应声明下面的可选特性（如果存在的话）哪些被支持：
  - 基于 objectClass 进行条目过滤；
  - 通过 AttributeSelection 进行属性的选择/排除；
  - 在复制域中包含下级知识；
  - 除下级知识外，还包含扩展的知识；
  - 基于上下文进行属性值的选择/排除。

#### 13.3.2 静态要求

一个 DSA 应当：

- a) 具备能力支持在第 7 章定义了抽象语法的应用上下文，以及在第 10 章定义 IDM 协议，这些内容的一致性已经被声明；

- b) 提供对modifyTimestamp 和createTimestamp 操作属性的支持。

### 13.3.3 动态要求

一个 DSA 应当:

- a) 如果对 8.2.3 定义的任何应用上下文声明了一致性,则应遵循在第 8 章定义的到已用 OSI 服务的映射;
- b) 遵循 ISO/IEC 9594-9:2005 中与 DISP 相关的规程。

### 13.4 影像消费者的一致性

一个声明遵循本目录规范的,并承担影像消费者角色的 DSA 实现,应当满足 13.4.1 到 13.4.3 规定的要求。

#### 13.4.1 声明要求

下面内容应当被声明:

- a) 对下列作为影像消费者的应用上下文进行一致性声明:shadowSupplierInitiatedAC, shadowConsumerInitiatedAC, shadowSupplierInitiatedAsynchronousAC, shadowConsumerInitiatedAsynchronousAC 和 disp-ip。

一个在一致性声明中声明为影像消费者,但不支持 disp-ip 的 DSA 实现,应当最少或者支持 shadowSupplierInitiatedAC, 或者支持 shadowConsumerInitiatedAC。如果该 DSA 支持 shadowSupplierInitiatedAC, 则它可以可选地支持 shadowSupplierInitiatedAsynchronousAC。如果该 DSA 支持 shadowConsumerInitiatedAC, 则它可以可选地支持 shadowConsumerInitiatedAsynchronousAC。如果声明了遵循 disp-ip, 则它应当被声明其实现是否有能力响应 requestShadowUpdate 操作, 或者有能力请求 coordinateShadowUpdate, 或者两者均可;

- b) 对安全级别进行一致性声明(无安全,简单安全,强安全);
- c) DSA 是否能够作为一个二次影像提供者(即作为一个中间 DSA,参与到二次影像中);
- d) DSA 是否支持对复制的重叠单元进行影像。

#### 13.4.2 静态要求

一个 DSA 应当:

- a) 具备能力支持在第 7 章定义了抽象语法的应用上下文,以及在第 10 章定义 IDM 协议,这些内容的一致性已经被声明;
- b) 如果支持复制的重叠单元,则提供对modifyTimestamp 和createTimestamp 操作属性的支持;
- c) 提供对copyShallDo 服务控制的支持。

#### 13.4.3 动态要求

一个 DSA 应当:

- a) 如果对任何应用上下文声明了一致性,则应遵循第 8 章定义的到已用 OSI 服务的映射;
- b) 遵循 ISO/IEC 9594-9:2005 中与 DISP 相关的规程。

附 录 A  
(规范性附录)

用 ASN.1 描述的公共协议规范

---

CommonProtocolSpecification {joint-iso-itu-t ds(5) module (1) commonProtocolSpecification (35) 5}

DEFINITIONS::=

BEGIN

--EXPORTS All --

——本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块，

——以及使用它们访问目录服务的其他应用。

——其他应用可把它们用于自己的目的，

——但这并不限制为维护或改进目录服务所需的扩充和修改。

IMPORTS

——来自 GB/T 16264.2—2008

    opBindingManagement

        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

    establishOperationalBinding, modifyOperationalBinding, terminateOperationalBinding

        FROM OperationalBindingManagement opBindingManagement;

OPERATION::= CLASS {

    &.ArgumentType     OPTIONAL,  
    &.ResultType       OPTIONAL,  
    &.Errors            ERROR OPTIONAL,  
    &.operationCode     Code UNIQUE OPTIONAL}

WITH SYNTAX {

    [ARGUMENT   &.ArgumentType]  
    [RESULT     &.ResultType]  
    [ERRORS     &.Errors]  
    [CODE       &.operationCode]}

ERROR::= CLASS {

    &.ParameterType,  
    &.errorCode     Code UNIQUE OPTIONAL}

WITH SYNTAX {

    PARAMETER   &.ParameterType  
    [CODE       &.errorCode]}

Code ::= CHOICE {

    local        INTEGER,  
    global       OBJECTIDENTIFIER}

Invokeld ::= CHOICE {

present INTEGER,

absent NULL}

—— 用于DAP和DSP的操作代码

id-opcode-read Code ::= local: 1

id-opcode-compare Code ::= local: 2

id-opcode-abandon Code ::= local: 3

id-opcode-list Code ::= local: 4

id-opcode-search Code ::= local: 5

id-opcode-addEntry Code ::= local: 6

id-opcode-removeEntry Code ::= local: 7

id-opcode-modifyEntry Code ::= local: 8

id-opcode-modifyDN Code ::= local: 9

—— 用于DISP的操作代码

id-opcode-requestShadowUpdate Code ::= local: 1

id-opcode-updateShadow Code ::= local: 2

id-opcode-coordinateShadowUpdate Code ::= local: 3

—— 用于DOP的操作代码

id-op-establishOperationalBinding Code ::= local: 100

id-op-modifyOperationalBinding Code ::= local: 102

id-op-terminateOperationalBinding Code ::= local: 101

—— 用于DAP和DSP的差错代码

id-errcode-attributeError Code ::= local: 1

id-errcode-nameError Code ::= local: 2

id-errcode-serviceError Code ::= local: 3

id-errcode-referral Code ::= local: 4

id-errcode-abandoned Code ::= local: 5

id-errcode-securityError Code ::= local: 6

id-errcode-abandonFailed Code ::= local: 7

id-errcode-updateError Code ::= local: 8

id-errcode-dsaReferral Code ::= local: 9

—— 用于DISP的差错代码

id-errcode-shadowError Code ::= local: 1

—— 用于DOP的差错代码

id-err-operationalBindingError Code ::= local: 100

DOP-Invokable OPERATION ::= {establishOperationalBinding |  
modifyOperationalBinding |  
terminateOperationalBinding}

DOP-Returnable OPERATION::= {establishOperationalBinding |  
modifyOperationalBinding |  
terminateOperationalBinding}

END--CommonProtocolSpecification

---

国家图书馆专用



**附 录 B**  
(规范性附录)

**用 ASN.1 描述的 OSI 协议**

OSIProtocolSpecification {joint-iso-itu-t ds(5) module (1) oSIProtocolSpecification (36) 5}

DEFINITIONS ::=

BEGIN

--EXPORTS All --

——本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块，

——以及使用它们访问目录服务的其他应用。

——其他应用可把它们用于自己的目的，

——但这并不限制为维护或改进目录服务所需的扩充和修改。

IMPORTS

——来自 GB/T 16264.2—2008

commonProtocolSpecification, directoryAbstractService, directoryOSIProtocols,  
enhancedSecurity, informationFramework

FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

Name, RelativeDistinguishedName

FROM InformationFramework informationFramework

OPTIONALLY-PROTECTED

FROM EnhancedSecurity enhancedSecurity

——来自 GB/T 16264.3—2008

SecurityProblem, ServiceProblem, Versions

FROM DirectoryAbstractService directoryAbstractService

——来自 GB/T 16264.5—2008

InvokeId, OPERATION

FROM CommonProtocolSpecification commonProtocolSpecification

APPLICATION-CONTEXT

FROM DirectoryOSIProtocols directoryOSIProtocols;

--OSI 协议 1--

OSI-PDU {APPLICATION-CONTEXT; protocol} ::= TYPE-IDENTIFIER. &.Type(

OsiBind {{protocol}}|

OsiBindResult {{protocol}}|

OsiBindError {{protocol}}|

OsiOperation {{protocol. &. Operations}}|

PresentationAbort)

OsiBind {APPLICATION-CONTEXT; Protocols} ::= SET {

mode-selector

[0] IMPLICIT SET {mode-value [0] IMPLICIT IN-

TEGER (1)},

normal-mode-parameters	[2] IMPLICIT SEQUENCE {
protocol-version	[0] IMPLICIT BIT STRING {version-1(0)} DEFAULT {version-1},
calling-presentation-selector	[1] IMPLICIT Presentation-selector OPTIONAL,
called-presentation-selector	[2] IMPLICIT Presentation-selector OPTIONAL,
presentation-context-definition-list	[4] IMPLICIT Context-list,
user-data	CHOICE {
fully-encoded-data	[APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SEQUENCE {
transfer-syntax-name	Transfer-syntax-name OPTIONAL,
presentation-context-identifier	Presentation-context-identifier,
presentation-data-values	CHOICE {
single-ASN1-type	[0] AARQ-apdu {{Protocols}}}}}

Presentation-selector ::= OCTET STRING(SIZE (1..4, ..., 5..MAX))

Context-list ::= SEQUENCE SIZE (2) OF

SEQUENCE {	
presentation-context-identifier	Presentation-context-identifier,
abstract-syntax-name	Abstract-syntax-name,
transfer-syntax-name-list	SEQUENCE OF Transfer-syntax-name}

Presentation-context-identifier ::= INTEGER(1..127, ..., 128..MAX)

Abstract-syntax-name ::= OBJECT IDENTIFIER

Transfer-syntax-name ::= OBJECT IDENTIFIER

AARQ-apdu { APPLICATION-CONTEXT; Protocols } ::= [APPLICATION 0] IMPLICIT SEQUENCE {

protocol-version	[0] IMPLICIT BIT STRING {version1(0)} DEFAULT {version1},
application-context-name	[1] Application-context-name,
called-AP-title	[2] Name OPTIONAL,
called-AE-qualifier	[3] RelativeDistinguishedName OPTIONAL,
called-AP-invocation-identifier	[4] AP-invocation-identifier OPTIONAL,
called-AE-invocation-identifier	[5] AE-invocation-identifier OPTIONAL,
calling-AP-title	[6] Name OPTIONAL,
calling-AE-qualifier	[7] RelativeDistinguishedName OPTIONAL,
calling-AP-invocation-identifier	[8] AP-invocation-identifier OPTIONAL,
calling-AE-invocation-identifier	[9] AE-invocation-identifier OPTIONAL,

implementation-information [29] IMPLICIT Implementation-data OPTIONAL,  
 user-information [30] IMPLICIT Association-informationBind {{Proto-  
 cols}}}

Association-informationBind {APPLICATION-CONTEXT:Protocols} ::= SEQUENCE SIZE(1) OF  
 EXTERNAL {

WITH COMPONENTS {  
 identification (WITH COMPONENTS {syntax ABSENT}),  
 data-value-descriptor ABSENT,  
 data-value (CONTAINING TheOsiBind {{Protocols}})}

Application-context-name ::= OBJECT IDENTIFIER

AP-invocation-identifier ::= INTEGER

AE-invocation-identifier ::= INTEGER

Implementation-data ::= GraphicString

TheOsiBind {APPLICATION-CONTEXT:Protocols} ::= [16] APPLICATION-CONTEXT. &bind-operation. &ArgumentType ({Protocols})

OsiBindResult {APPLICATION-CONTEXT:Protocols} ::= SET {  
 mode-selector [0] IMPLICIT SET {mode-value [0] IMPLICIT INTEGER  
 (1)},  
 normal-modeparameters [2] IMPLICIT SEQUENCE {  
 protocol-version [0] IMPLICIT BIT STRING {version-1(0)} DE-  
 FAULT {version-1},  
 responding-presentation-selector  
 [3] IMPLICIT Presentation-selector OPTIONAL,  
 presentation-context-definition-result-list  
 [5] IMPLICIT SEQUENCE SIZE (2) OF SEQUENCE {  
 result [0] IMPLICIT Result (acceptance),  
 transfer-syntax-name [1] IMPLICIT Transfer-syntax-name),  
 user-data CHOICE {  
 fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE  
 (1) OF SEQUENCE {  
 transfer-syntax-name Transfer-syntax-name OPTIONAL,  
 presentation-context-identifier Presentation-context-identifier,  
 presentation-data-values CHOICE {  
 single-ASN1-type [0] AARE-apdu {{Protocols}}}}}

Result ::= INTEGER {

acceptance (0),  
 user-rejection (1),  
 provider-rejection (2)}

AARE-apdu { APPLICATION-CONTEXT; Protocols } ::= [ APPLICATION 1 ] IMPLICIT SEQUENCE {

protocol-version [0]  
     IMPLICIT BIT STRING {version1(0)} DEFAULT {version1},  
 application-context-name [1] Application-context-name,  
 result [2] Associate-result (accepted),  
 result-source-diagnostic [3] Associate-source-diagnostic,  
 responding-AP-title [4] Name OPTIONAL,  
 responding-AE-qualifier [5] RelativeDistinguishedName OPTIONAL,  
 responding-AP-invocation-identifier [6] AP-invocation-identifier OPTIONAL,  
 responding-AE-invocation-identifier [7] AE-invocation-identifier OPTIONAL,  
 implementation-information [29] IMPLICIT Implementation-data OPTIONAL,  
 user-information [30] IMPLICIT Association-informationBindRes {{Protocols}}

Association-informationBindRes { APPLICATION-CONTEXT; Protocols } ::= SEQUENCE SIZE(1)  
 OF EXTERNAL (

WITH COMPONENTS {  
     identification (WITH COMPONENTS {syntax ABSENT}),  
     data-value-descriptor ABSENT,  
     data-value (CONTAINING TheOsBindRes {{Protocols}})}

Associate-result ::= INTEGER {

accepted (0),  
 rejected-permanent (1),  
 rejected-transient (2)} {0..2,...}

Associate-source-diagnostic ::= CHOICE {

acse-service-user [1] INTEGER {  
     null (0),  
     no-reason-give (1),  
     application-context-name-not-supported (2),  
     calling-AP-title-not-recognized (3),  
     calling-AP-invocation-identifier-not-recognized (4),  
     calling-AE-qualifier-not-recognized (5),  
     calling-AE-invocation-identifier-not-recognized (6),  
     called-AP-title-not-recognized (7),  
     called-AP-invocation-identifier-not-recognized (8),  
     called-AE-qualifier-not-recognized (9),  
     called-AE-invocation-identifier-not-recognized (10)} {0..10,...},

acse-service-provider [2] INTEGER {  
     null (0),  
     no-reason-given (1),  
     no-common-acse-version (2)} (0..2, ...)

TheOsiBindRes {APPLICATION-CONTEXT;Protocols} ::=

17] APPLICATION-CONTEXT, & bind-operation, & ResultType({Protocols})

OsiBindError{APPLICATION-CONTEXT;Protocols} ::= CHOICE {

normal-mode-parameters SEQUENCE {

    protocol-version [0] IMPLICIT BIT STRING {version-1(0)} DEFAULT{version-1},

    responding-presentation-selector

        [3] IMPLICIT Presentation-selector OPTIONAL,

    presentation-context-definition-result-list

        [5] IMPLICIT Result-list OPTIONAL,

    provider-reason [10] IMPLICIT Provider-reason OPTIONAL,

    user-data CHOICE {

        fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1)

OF SEQUENCE {

        transfer-syntax-name Transfer-syntax-name OPTIONAL,

        presentation-context-identifier Presentation-context-identifier,

        presentation-data-values CHOICE {

            single-ASN1-type [0] AARErr-apdu {{Protocols}}}}

OPTIONAL}}

AARErr-apdu (APPLICATION-CONTEXT;Protocols) ::= [APPLICATION 1] IMPLICIT SEQUENCE {

    protocol-version [0] IMPLICIT BIT STRING {version1(0)}

        DEFAULT {version1},

    application-context-name [1] Application-context-name,

    result [2] Associate-result (rejected-permanent..rejected-transient),

    result-source-diagnostic [3] Associate-source-diagnostic,

    responding-AP-title [4] Name OPTIONAL,

    responding-AE-qualifier [5] RelativeDistinguishedName OPTIONAL,

    responding-AP-invocation-identifier [6] AP-invocation-identifier OPTIONAL,

    responding-AE-invocation-identifier [7] AE-invocation-identifier OPTIONAL,

    implementation-information [29] IMPLICIT Implementation-data OPTIONAL,

    user-information [30]

        IMPLICIT Association-informationBindErr {{Protocols}} OPTIONAL}

Association-informationBindErr {APPLICATION-CONTEXT;Protocols} ::= SEQUENCE SIZE(1) OF EXTERNAL (

    WITH COMPONENTS {

        identification (WITH COMPONENTS {syntax ABSENT}),

        data-value-descriptor ABSENT,

data-value (CONTAINING TheOsiBindErr {{Protocols}}))

TheOsiBindErr {APPLICATION-CONTEXT;Protocols} ::=  
 [18] APPLICATION-CONTEXT. &bind-operation. &Errors. &ParameterType  
 ({Protocols})

Result-list ::= SEQUENCE SIZE (2) OF SEQUENCE {  
 result [0] IMPLICIT Result,  
 transfer-syntax-name [1] IMPLICIT Transfer-syntax-name OPTIONAL,  
 provider-reason [2] IMPLICIT INTEGER {  
   reason-not-specified (0),  
   abstract-syntax-not-supported (1),  
   proposed-transfer-syntaxes-not-supported (2)} OPTIONAL}

Provider-reason ::= INTEGER {  
   reason-not-specified (0),  
   temporary-congestion (1),  
   local-limit-exceeded (2),  
   called-presentation-address-unknown (3),  
   protocol-version-not-supported (4),  
   default-context-not-supported (5),  
   user-data-not-readable (6),  
   no-PSAP-available (7)}

OsiUnbind ::= CHOICE {  
   fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SE-  
   QUENCE {  
     presentation-context-identifier Presentation-context-identifier,  
     presentation-data-values CHOICE {  
       single-ASN1-type [0] The OsiUnbind}}}

TheOsiUnbind ::= [APPLICATION 2] IMPLICIT SEQUENCE {  
   reason [0] IMPLICIT Release-request-reason OPTIONAL}

Release-request-reason ::= INTEGER {  
   normal (0)}

OsiUnbindResult ::= CHOICE {  
   fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SE-  
   QUENCE {  
     presentation-context-identifier Presentation-context-identifier,  
     presentation-data-values CHOICE {  
       single-ASN1-type [0] The OsiUnbindRes}}}

TheOsiUnbindRes ::= [APPLICATION 3] IMPLICIT SEQUENCE {  
     reason [0] IMPLICIT Release-response-reason OPTIONAL}

Release-response-reason ::= INTEGER {  
     normal (0)}

OsiOperation {OPERATION:Operations} ::= CHOICE {  
     fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SE-  
     QUENCE {  
         presentation-context-identifier Presentation-context-identifier,  
         presentation-data-values CHOICE {  
             single-ASN1-type [0] CHOICE {  
                 request OsiReq {{Operations}},  
                 result OsiRes {{Operations}},  
                 error OsiErr {{Operations}},  
                 reject OsiRej}}}}}

OsiReq {OPERATION:Operations} ::= [1] IMPLICIT SEQUENCE {  
     invokeld Invokeld,  
     opcode OPERATION. &operationCode ({Operations}),  
     argument OPERATION. &ArgumentType ({Operations} {@opcode})}

OsiRes {OPERATION:Operations} ::= [2] IMPLICIT SEQUENCE {  
     invokeld Invokeld,  
     result SEQUENCE {  
         opcode OPERATION. &operationCode ({Operations}),  
         result OPERATION. &ResultType ({Operations} {@opcode})}}

OsiErr {OPERATION:Operations} ::= [3] IMPLICIT SEQUENCE {  
     invokeld Invokeld,  
     errcode OPERATION. &Errors. &errorCode ({Operations}),  
     error OPERATION. &Errors. &ParameterType ({Operations} {@.errcode})}

OsiRej ::= [4] IMPLICIT SEQUENCE {  
     invokeld Invokeld,  
     problem CHOICE {  
         general [0] GeneralProblem,  
         invoke [1] InvokeProblem,  
         returnResult [2] ReturnResultProblem,  
         returnError [3] ReturnErrorProblem}}

GeneralProblem ::= INTEGER {  
     unrecognizedPDU (0),

mistypedPDU (1),  
badlyStructuredPDU (2)}

InvokeProblem ::= INTEGER {  
duplicateInvocation (0),  
unrecognizedOperation (1),  
mistypedArgument (2),  
resourceLimitation (3),  
releaseInProgress (4)}

ReturnResultProblem ::= INTEGER {  
unrecognizedInvocation (0),  
resultResponseUnexpected (1),  
mistypedResult (2)}

ReturnErrorProblem ::= INTEGER {  
unrecognizedInvocation (0),  
errorResponseUnexpected (1),  
unrecognizedError (2),  
unexpectedError (3),  
mistypedParameter (4)}

PresentationAbort ::= CHOICE {  
aru-ppdu ARU-PPDU,  
arp-ppdu ARP-PPDU}

ARU-PPDU ::= CHOICE {  
normal-mode-parameters [0] IMPLICIT SEQUENCE {  
presentation-context-identifier-list [0] IMPLICIT Presentation-context-identifier-list,  
user-data CHOICE {  
fully-encoded-data [APPLICATION 1] IMPLICIT SEQUENCE SIZE (1) OF SE-  
QUENCE {  
presentation-context-identifier Presentation-context-identifier,  
presentation-data-values CHOICE {  
single-ASN1-type [0] ABRT-apdu}}}}}

Presentation-context-identifier-list ::=  
SEQUENCE SIZE (1) OF SEQUENCE {  
presentation-context-identifier Presentation-context-identifier,  
transfer-syntax-name Transfer-syntax-name}

ABRT-apdu ::= [APPLICATION 4] IMPLICIT SEQUENCE {  
abort-source ABRT-source}



ABRT-source ::= INTEGER {  
     acse-service-user       (0),  
     acse-service-provider (1)}

ARP-PPDU ::= SEQUENCE {  
     provider-reason     [0] IMPLICIT Abort-reason OPTIONAL,  
     event-identifier    [1] IMPLICIT Event-identifier OPTIONAL}

Abort-reason ::= INTEGER {  
     reason-not-specified               (0),  
     unrecognized-ppdu                 (1),  
     unexpected-ppdu                   (2),  
     unexpected-session-service-primitive (3),  
     unrecognized-ppdu-parameter       (4),  
     unexpected-ppdu-parameter        (5),  
     invalid-ppdu-parameter-value      (6)}

Event-identifier ::= INTEGER {  
     cp-PPDU                   (0),  
     cpa-PPDU                  (1),  
     cpr-PPDU                  (2),  
     aru-PPDU                  (3),  
     arp-PPDU                  (4),  
     td-PPDU                   (7),  
     s-release-indication       (14),  
     s-release-confirm          (15)}

END--OSIPProtocolSpecification

---

附 录 C  
(规范性附录)

用 ASN.1 描述的目录 OSI 协议

DirectoryOSIProtocols {joint-iso-itu-t ds(5)module(1) directoryOSIProtocols(37)5}

DEFINITIONS::=

BEGIN

--EXPORTS All --

——本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块，

——以及使用它们访问目录服务的其他应用。

——其他应用可把它们用于自己的目的，

——但这并不限制为维护或改进目录服务所需的扩充和修改。

IMPORTS

——来自GB/T 16264.2—2008

commonProtocolSpecification, directoryAbstractService, distributedOperations,  
directoryShadowAbstractService, id-ac, id-as, id-idm, idMProtocolSpecification,  
opBindingManagement, oSIProtocolSpecification

FROM UsefulDefinitions {joint-iso-itu-t ds(5)module(1) usefulDefinitions(0) 5}

dSAOperationalBindingManagementBind, establishOperationalBinding, modifyOperational-  
Binding,

terminateOperationalBinding

FROM OperationalBindingManagement opBindingManagement

——来自GB/T 16264.3—2008

abandon, addEntry, compare, directoryBind, list, modifyDN, modifyEntry, read, re-  
moveEntry, search

FROM DirectoryAbstractService directoryAbstractService

——来自GB/T 16264.4—2008

chainedAbandon, chainedAddEntry, chainedCompare, chainedList, chainedModifyDN,  
chainedModifyEntry, chainedRead, chainedRemoveEntry, chainedSearch, dSABind

FROM DistributedOperations distributedOperations

——来自GB/T 16264.5—2008

OPERATION

FROM Common ProtocolSpecification commonProtocolSpecification

OSI-PDU {}

FROM OSIProtocolSpecifications oSIProtocolSpecification

——来自ISO/IEC 9594-9:2005

coordinateShadowUpdate, dSAShadowBind, requestShadowUpdate, updateShadow

FROM DirectoryShadowAbstractService directoryShadowAbstractService;

--OSI protocols--

DAP-OSI-PDUs ::= OSI-PDU {directoryAccessAC}

DSP-OSI-PDUs ::= OSI-PDU {directorySystemAC}

DOP-OSI-PDUs ::= OSI-PDU {directoryOperationalBindingManagementAC}

ShadowSupplierInitiatedDISP-OSI-PDUs ::= OSI-PDU {shadowSupplierInitiatedAC}

ShadowSupplierInitiatedAsynchronousDISP-OSI-PDUs ::=  
OSI-PDU {shadowSupplierInitiatedAsynchronousAC}

ShadowConsumerInitiatedDISP-OSI-PDUs ::= OSI-PDU {shadowConsumerInitiatedAC}

ShadowConsumerInitiatedAsynchronousDISP-OSI-PDUs ::=  
OSI-PDU {shadowConsumerInitiatedAsynchronousAC}

APPLICATION-CONTEXT ::= CLASS {  
    &.bind-operation                      OPERATION;  
    &.Operations                          OPERATION;  
    &.applicationContextName            OBJECT IDENTIFIER UNIQUE}  
WITHSYNTAX {  
    BIND-OPERATION                      &.bind-operation  
    OPERATIONS                          &.Operations  
    APPLICATION CONTEXT NAME &.applicationContextName}

directoryAccessAC APPLICATION-CONTEXT ::= {  
    BIND-OPERATION                      directoryBind  
    OPERATIONS                          {read | compare | abandon | list | search  
   | addEntry | removeEntry | modifyEntry | modifyDN}  
    APPLICATION CONTEXT NAME id-ac-directoryAccessAC}

directorySystemAC APPLICATION-CONTEXT ::= {  
    BIND-OPERATION                      dSABind  
    OPERATIONS                          {chainedRead | chainedCompare | chainedAbandon  
   | chainedList | chainedSearch  
   | chainedAddEntry | chainedRemoveEntry  
   | chainedModifyEntry | chainedModifyDN}  
    APPLICATION CONTEXTNAME id-ac-directorySystemAC}

shadowSupplierInitiatedAC APPLICATION-CONTEXT ::= {  
    BIND-OPERATION                      dSAShadowBind  
    OPERATIONS                          {updateShadow}

```

        | coordinateShadowUpdate}
APPLICATION CONTEXT NAME id-ac-shadowSupplierInitiatedAC}

shadowConsumerInitiatedAC APPLICATION-CONTEXT ::= {
    BIND-OPERATION          dSAShadowBind
    OPERATIONS              { requestShadowUpdate
                            | updateShadow}
APPLICATION CONTEXT NAME id-ac-shadowConsumerInitiatedAC}

shadowSupplierInitiatedAsynchronousAC APPLICATION-CONTEXT ::= {
    BIND-OPERATION          dSAShadowBind
    OPERATIONS              { updateShadow
                            | coordinateShadowUpdate}
APPLICATION CONTEXT NAME id-ac-shadowSupplierInitiatedAsynchronousAC}

shadowConsumerInitiatedAsynchronousAC APPLICATION-CONTEXT ::= {
    BIND-OPERATION          dSAShadowBind
    OPERATIONS              { requestShadowUpdate
                            | updateShadow}
APPLICATION CONTEXT NAME id-ac-shadowConsumerInitiatedAsynchronousAC}

directoryOperationalBindingManagementAC APPLICATION-CONTEXT ::= {
    BIND-OPERATION          dSAOperationalBindingManagementBind
    OPERATIONS              { establishOperationalBinding
                            | modifyOperationalBinding
                            | terminateOperationalBinding}
APPLICATION CONTEXT NAME id-ac-directoryOperationalBinding ManagementAC}

--抽象语法--
id-as-directoryAccessAS          OBJECT IDENTIFIER ::= {id-as 1}
id-as-directorySystemAS          OBJECT IDENTIFIER ::= {id-as 2}
id-as-directoryShadowAS          OBJECT IDENTIFIER ::= {id-as 3}
id-as-directoryOperationalBindingManagementAS OBJECT IDENTIFIER ::= {id-as 4}
--id-as-directoryReliableShadowAS OBJECT IDENTIFIER ::= {id-as 5}
--id-as-reliableShadowBindingAS   OBJECT IDENTIFIER ::= {id-as 6}
--id-as-2or3se                   OBJECT IDENTIFIER ::= {id-as 7}

id-acseAS                        OBJECT IDENTIFIER ::=

--application context object identifiers

id-ac-directoryAccessAC          OBJECT IDENTIFIER ::= {id-ac 1}
id-ac-directorySystemAC          OBJECT IDENTIFIER ::= {id-ac 2}
id-ac-directoryOperationalBindingManagementAC OBJECT IDENTIFIER ::= {id-ac 3}

```

id-ac-shadowConsumerInitiatedAC	OBJECT IDENTIFIER ::= {id-ac 4}
id-ac-shadowSupplierInitiatedAC	OBJECT IDENTIFIER ::= {id-ac 5}
--id-ac-reliableShadowSupplierInitiatedAC	OBJECT IDENTIFIER ::= {id-ac 6}
--id-ac-reliableShadowConsumerInitiatedAC	OBJECT IDENTIFIER ::= {id-ac 7}
id-ac-shadowSupplierInitiatedAsynchronousAC	OBJECT IDENTIFIER ::= {id-ac 8}
id-ac-shadowConsumerInitiatedAsynchronousAC	OBJECT IDENTIFIER ::= {id-ac 9}
--id-ac-directoryAccessWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 10}
--id-ac-directorySystemWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 11}
--id-ac-shadowSupplierInitiatedWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 12}
--id-ac-shadowConsumerInitiatedWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 13}
--id-ac-reliableShadowSupplierInitiatedWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 14}
--id-ac-reliableShadowConsumerInitiatedWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 15}
--id-ac-directoryOperationalBindingManagementWith2or3seAC	OBJECT IDENTIFIER ::= {id-ac 16}

END--DirectoryOSIProtocols

国家图书馆专用

附 录 D  
(规范性附录)  
用 ASN.1 描述 IDM 协议

本附录包含了本目录规范中的所有 ASN.1 类型和值定义,以 ASN.1 模块IDMProtocolSpecification 的形式描述。

---

```
IDMProtocolSpecification {joint-iso-itu-t ds(5)module (1)iDMProtocolSpecification (30)5}
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
——本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块,
——以及使用它们访问目录服务的其他应用。
——其他应用可把它们用于自己的目的,
——但这并不限制为维护或改进目录服务所需的扩充和修改。
IMPORTS
—— 来自GB/T 16264.2—2008
    certificateExtensions, commonProtocolSpecification, directoryAbstractService, directory-
IDMProtocols,
    enhancedSecurity
    FROM UsefulDefinitions {joint-iso-itu-t ds(5)module(1) usefulDefinitions(0)5}
—— 来自ISO/IEC 9594-8:2005
    GeneralName
    FROM CertificateExtensions certificateExtensions
—— 来自GB/T 16264.3—2008
    SecurityProblem, ServiceProblem, Versions
    FROM DirectoryAbstractService directoryAbstractService
—— 来自GB/T 16264.5—2008
    InvokeId, OPERATION
    FROM CommonProtocolSpecification commonProtocolSpecification ;
—— IDM 协议信息客体类 ——
IDM-PROTOCOL ::= CLASS {
    &.bind-operation      OPERATION,
    &.Operations          OPERATION,
    &.id                  OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
    BIND-OPERATION      &.bind-operation
    OPERATIONS          &.Operations
    ID                   &.id }
-- IDM 协议 --
IDM-PDU {IDM-PROTOCOL:protocol} ::= CHOICE {
    bind                [0]      IdmBind{ {protocol} },
```

bindResult	[1]	IdmBindResult{ {protocol} },
bindError	[2]	IdmBindError{ {protocol} },
request	[3]	Request{ {protocol. &Operations} },
result	[4]	IdmResult{ {protocol. &Operations} },
error	[5]	Error{ {protocol. &Operations} },
reject	[6]	IdmReject,
unbind	[7]	Unbind,
abort	[8]	Abort,
startTLS	[9]	StartTLS,
tLSResponse	[10]	TLSResponse }

IdmBind {IDM-PROTOCOL;Protocols} ::= SEQUENCE {

protocolID		IDM-PROTOCOL. &id ( {Protocols} ),
callingAETitle	[0]	GeneralName OPTIONAL,
calledAETitle	[1]	GeneralName OPTIONAL,
argument	[2]	IDM-PROTOCOL. &bind-operation. &ArgumentType ( {Protocols} { @protocolID} ) }

IdmBindResult {IDM-PROTOCOL;Protocols} ::= SEQUENCE {

protocolID		IDM-PROTOCOL. &id ( {Protocols} ),
respondingAETitle	[0]	GeneralName OPTIONAL,
result	[1]	IDM-PROTOCOL. &bind-operation. &ResultType ( {Protocols} { @protocolID} ) }

IdmBindError {IDM-PROTOCOL;Protocols} ::= SEQUENCE {

protocolID		IDM-PROTOCOL. &id ( {Protocols} ),
errcode		IDM-PROTOCOL. &bind-operation. &Errors. &errorCode ( {Protocols} { @protocolID} ),
respondingAETitle	[0]	GeneralName OPTIONAL,
aETitleError		ENUMERATED { callingAETitleNotAccepted (0), calledAETitleNotRecognized (1) } OPTIONAL,
error	[1]	IDM-PROTOCOL. &bind-operation. &Errors. &ParameterType ( {Protocols} { @protocolID, @errcode} ) }

Unbind ::= NULL

Request {OPERATION;Operations} ::= SEQUENCE {

invokeID		INTEGER,
opcode		OPERATION. &operationCode ( {Operations} ),
argument		OPERATION. &ArgumentType ( {Operations} { @opcode} ) }

IdmResult {OPERATION;Operations} ::= SEQUENCE {

invokeID		InvokeId,
opcode		OPERATION. &operationCode ( {Operations} ),
result		OPERATION. &ResultType ( {Operations} { @opcode} ) }

Error {OPERATION;Operations} ::= SEQUENCE {

invokeID		INTEGER,
errcode		OPERATION. &Errors. &errorCode ( {Operations} ),

error                      OPERATION, &Errors, &ParameterType  
                               ({Operations} {@errcode}) }

IdmReject ::= SEQUENCE {  
     invokeID INTEGER,  
     reason ENUMERATED {  
         mistypedPDU                      (0),  
         duplicateInvokeIDRequest        (1),  
         unsupportedOperationRequest     (2),  
         unknownOperationRequest        (3),  
         mistypedArgumentRequest        (4),  
         resourceLimitationRequest       (5),  
         unknownInvokeIDResult          (6),  
         mistypedResultRequest          (7),  
         unknownInvokeIDError           (8),  
         unknownError                    (9),  
         mistypedParameterError        (10) } }

Abort ::= ENUMERATED {  
     mistypedPDU                      (0),  
     unboundRequest                    (1),  
     invalidPDU                        (2),  
     resourceLimitation                (3),  
     connectionFailed                  (4),  
     invalidProtocol                    (5),  
     reasonNotSpecified                (6) }

StartTLS ::= NULL

TLSResponse ::= ENUMERATED {  
     success                            (0),  
     operationsError                    (1),  
     protocolError                      (2),  
     unavailable                        (3) }

END --IDMProtocolSpecification

---



附 录 E  
(规范性附录)

用 ASN.1 描述的目录 IDM 协议

本附录包含了本目录规范中的所有相关的 ASN.1 类型和值定义,以 ASN.1 模块 DirectoryIDMP-rotocols 的形式描述。

---

```

DirectoryIDMPProtocols {joint-iso-itu-t ds(5) module(1) directoryIDMPProtocols(31) 5}
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
——本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块,
——以及使用它们访问目录服务的其他应用。
——其他应用可把它们用于自己的目的,
——但这并不限制为维护或改进目录服务所需的扩充和修改。
IMPORTS
—— 来自GB/T 16264.2—2008
    directoryAbstractService, distributedOperations, directoryShadowAbstractService, id-idm,
    iDMProtocolSpecification, opBindingManagement
    FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

    establish OperationalBinding, modifyOperationalBinding, terminateOperationalBinding
    FROM OperationalBindingManagement opBindingManagement
—— 来自GB/T 16264.3—2008
    abandon, addEntry, compare, directoryBind, list, modifyDN, modifyEntry, read, re-
moveEntry, search
    FROM DirectoryAbstractService directoryAbstractService
—— 来自GB/T 16264.4—2008
    chainedAbandon, chainedAddEntry, chainedCompare, chainedList, chainedModifyDN,
    chainedModifyEntry, chainedRead, chainedRemoveEntry, chainedSearch
    FROM DistributedOperations distributedOperations
—— 来自GB/T 16264.5—2008
    IDM-PDU, IDM-PROTOCOL
    FROM iDMProtocolSpecification iDMProtocolSpecification
—— 来自ISO/IEC 9594-9:2005
    coordinateShadowUpdate, requestShadowUpdate, updateShadow
    FROM DirectoryShadowAbstractService directoryShadowAbstractService;
-- IDM 协议--
DAP-IDM-PDUs ::= IDM-PDU {dap-ip}

dap-ip IDM-PROTOCOL ::= {
    BIND-OPERATION    directoryBind

```

OPERATIONS	{read   compare   abandon   list   search   addEntry   removeEntry   modifyEntry   modifyDN}
ID	id-idm-dap}

DSP-IDM-PDUs:: = IDM-PDU {dsp-ip}

dsp-ipIDM-PROTOCOL:: = {

BIND-OPERATION	directoryBind
OPERATIONS	{chainedRead   chainedCompare   chainedAbandon   chainedList   chainedSearch   chainedAddEntry   chainedRemoveEntry   chainedModifyEntry   chainedModifyDN}
ID	id-idm-dsp}

DISP-IDM-PDUs:: = IDM-PDU {disp-ip}

disp-ipIDM-PROTOCOL:: = {

BIND-OPERATION	directoryBind
OPERATIONS	{requestShadowUpdate   updateShadow   coordinateShadowUpdate}
ID	id-idm-disp}

DOP-IDM-PDUs:: = IDM-PDU {dop-ip}

dop-ipIDM-PROTOCOL:: = {

BIND-OPERATION	directoryBind
OPERATIONS	{establishOperationalBinding   modifyOperationalBinding   terminateOperationalBinding}
ID	id-idm-dop}

--协议客体标识符--

id-idm-dap	OBJECTIDENTIFIER	::=	{id-idm 0}
id-idm-dsp	OBJECTIDENTIFIER	::=	{id-idm 1}
id-idm-disp	OBJECTIDENTIFIER	::=	{id-idm 2}
id-idm-dop	OBJECTIDENTIFIER	::=	{id-idm 3}

END--DirectoryIDMProtocols

附 录 F  
(规范性附录)  
目录操作绑定类型

本附录包含了为标识本系列目录规范中的操作绑定类型而分配的所有 ASN.1 客体标识符,以 ASN.1 模块DirectoryOperationalBindingTypes 的形式提供。

---

---

DirectoryOperationalBindingTypes

{joint-iso-itu-t ds(5) module (1) directoryOperationalBindingTypes(25) 5 }

DEFINITIONS ::=

BEGIN

--EXPORTS All --

——本模块中定义的类型和值输出可用于目录规范包含的其他 ASN.1 模块,

——以及使用它们访问目录服务的其他应用。

——其他应用可把它们用于自己的目的,

——但这并不限制为维护或改进目录服务所需的扩充和修改。

IMPORTS

——来自GB/T 16264.2—2008

id-ob

FROM UsefulDefinitions{joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5} ;

id-op-binding-shadow OBJECT IDENTIFIER ::= { id-ob 1 }

id-op-binding-hierarchical OBJECT IDENTIFIER ::= { id-ob 2 }

id-op-binding-non-specific-hierarchical OBJECT IDENTIFIER ::= { id-ob 3 }

END --DirectoryOperationalBindingTypes

---

---

\_\_\_\_\_

中 华 人 民 共 和 国  
国 家 标 准  
信息技术 开放系统互连 目录  
第 5 部分:协议规范

GB/T 16264.5—2008/ISO/IEC 9594-5:2005

\*

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

网址 [www.spc.net.cn](http://www.spc.net.cn)

电话:68523946 68517548

中国标准出版社秦皇岛印刷厂印刷  
各地新华书店经销

\*

开本 880×1230 1/16 印张 5.25 字数 150 千字  
2008年12月第一版 2008年12月第一次印刷

\*

书号: 155066·1-34755

如有印装差错 由本社发行中心调换  
版权专有 侵权必究  
举报电话:(010)68533533



GB/T 16264.5-2008