



中华人民共和国国家标准

GB/T 16645.1—1996
idt ISO/IEC 9596-1:1991

信息技术 开放系统互连 公共管理信息协议 第1部分:规范

Information technology—Open System
Interconnection—Common management information
protocol—Part 1: Specification

1996-12-17 发布

1997-07-01 实施

国家技术监督局 发布

目 次

前言	Ⅲ
ISO/IEC 前言	Ⅳ
1 范围	1
2 引用标准	1
3 定义	2
3.1 基本参考模型定义	2
3.2 管理框架定义	2
3.3 远程操作定义	2
3.4 CMIS 定义	2
3.5 ACSE 定义	2
3.6 表示定义	3
4 符号和缩略语	3
5 概述	3
5.1 提供的服务	3
5.2 基础的服务	3
5.3 管理信息定义	4
5.4 协议版本	4
6 规程要素	4
6.1 联系建立	4
6.2 远程操作	4
6.3 事件报告规程	5
6.4 获得规程	5
6.5 设置规程	7
6.6 动作规程	7
6.7 建立规程	8
6.8 删除规程	9
6.9 联系有序释放	9
6.10 联系突然释放	9
7 抽象语法	9
7.1 约定	9
7.2 CMISE 原语和 CMIP 操作之间的对应关系	10
7.3 ACSE 用户数据	10
7.4 CMIP 数据单元	11
7.5 CMIP 抽象语法定义	23
8 一致性	23
8.1 静态要求	23

8.2 动态要求.....	24
附录 A(标准的附录) 用于 CMISE 的联系规则	25
附录 B(提示的附录) 扩充 ASN.1 语法.....	27
附录 C(提示的附录) CMISE ROSE APDU 例子	39

前 言

本标准等同采用国际标准 ISO/IEC 9596-1:1991《信息技术 开放系统互连 公共管理信息协议 第1部分:规范》,并根据国际标准 ISO/IEC 9596-1:1991/Cor. 1:1992《技术更正 1》、ISO/IEC 9596-1:1991/Cor. 2:1992《技术更正 2》、ISO/IEC 9595:1991/Cor. 3:1992《技术更正 3》、ISO/IEC 9596-1:1991/Cor. 4:1993《技术更正 4》和 ISO/IEC 9596-1:1991/Cor. 5:1994《技术更正 5》进行了相应修改:

- a) 根据《技术更正 1》修改章条号为:增加 5.4 和修改附录 C;
- b) 根据《技术更正 2》修改章条号为:5.3 和 6.10;
- c) 根据《技术更正 3》修改章条号为:第 2 章、8.1 和 8.2;
- d) 根据《技术更正 4》修改章条号为:7.4 和附录 B;
- e) 根据《技术更正 5》修改章条号为:7.4。

制定本国家标准,有利于我国开放系统互连的基本发展。

本标准与 ISO/IEC 9596-1:1991 的区别如下:

ISO/IEC 9596-1:1991 中,附录 D 是 CMIP 的一些修正,由于这些修正已合并在本标准中,所以将 ISO/IEC 9596-1:1991 中的附录 D 删去。

本标准的附录 A 是标准的附录;附录 B 和附录 C 是提示的附录。

本标准由中华人民共和国电子工业部提出。

本标准由电子工业部标准化研究所归口。

本标准起草单位:中国电子标准化研究所。

本标准主要起草人:高健、黄家英、冯惠。

ISO/IEC 前言

ISO(国际标准化组织)和 IEC(国际电工委员会)是世界性标准化专门机构。国家成员体(它们都是 ISO 或 IEC 的成员国)通过国际组织建立的各个技术委员会参与制定针对特定技术范围的国际标准。ISO 和 IEC 的各技术委员会在共同感兴趣的领域内进行合作。与 ISO 和 IEC 有联系的其他官方和非官方国际组织也可参与国际标准的制定工作。

对于信息技术,ISO 和 IEC 建立了一个联合技术委员会,即 ISO/IEC JTC1。由联合技术委员会提出的国际标准草案需分发给国家成员体进行表决。发布一项国际标准,至少需要 75%的参与表决的国家成员体投票赞成。

国际标准 ISO/IEC 9596-1 是由 ISO/IEC JTC1“信息技术”联合技术委员会制定的。

本系列标准的这一部分的附录 A 是标准的附录,附录 B 和附录 C 是提示的附录。

中华人民共和国国家标准

信息技术 开放系统互连 公共管理信息协议 第1部分:规范

GB/T 16645.1—1996
idt ISO/IEC 9596-1:1991

Information technology—Open System
Interconnection—Common management information
protocol—Part 1: Specification

1 范围

本系列标准的这一部分规定了由应用层实体使用的用来交换管理信息的协议。

本系列标准的这一部分规定:

- 在应用实体间传输管理信息的规程;
- 所使用的公共管理信息协议的抽象语法和相关编码规则;
- 正确解释协议控制信息的规程;
- 实现这一部分要满足的一致性要求。

本系列标准的这一部分不规定:

- 借助 CMIP 发送的管理信息的结构或意义;
- 作为 CMIP 交换结果来完成管理的方法;
- 导致 CMIP 使用的相互作用。

2 引用标准

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

GB 9387—88 信息处理系统 开放系统互连 基本参考模型(idt ISO 7498:1984)

GB/T 9387.4—1996 信息处理系统 开放系统互连 基本参考模型 第4部分:管理框架
(idt ISO 7498-4:1989)

GB/T 15128—94 信息处理系统 开放系统互连 面向连接的基本会话服务定义
(idt ISO 8326:1987)

GB/T 16688—1996 信息处理系统 开放系统互连 联系控制服务元素的服务定义
(idt ISO 8649:1987)

GB/T 16687—1996 信息处理系统 开放系统互连 联系控制服务元素的协议规范
(idt ISO 8650:1987)

GB/T 15695—1995 信息处理系统 开放系统互连 面向连接的表示服务定义
(idt ISO 8822:1987)

GB/T 15696—1995 信息处理系统 开放系统互连 面向连接的表示协议规范(idt ISO 8823:
1987)

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

(idt ISO/IEC 8824:1990)

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

GB/T 16644—1996 信息技术 开放系统互连 公共管理信息服务定义
(idt ISO/IEC 9595:1991)

ISO/IEC 9072-1:1989 信息处理系统 文本通信 远程操作 第1部分:模型、记法和服务定义

ISO/IEC 9072-2:1989 信息处理系统 文本通信 远程操作 第2部分:协议规范

3 定义

下列定义适用于本系列标准的这一部分。

3.1 基本参考模型定义

本系列标准的这一部分使用在 GB 9387 中定义的下列术语:

- a) 应用服务元素;
- b) 应用进程;
- c) 实开放系统;
- d) 系统管理。

3.2 管理框架定义

本系列标准的这一部分使用在 GB/T 9387.4 中定义的下列术语:

- a) 受管理客体;
- b) 管理信息;
- c) 管理信息库;
- d) 系统管理应用实体。

3.3 远程操作定义

本系列标准的这一部分使用在 ISO/IEC 9072:1989 中定义的下列术语:

- a) 联系启动者;
- b) 联系响应者;
- c) 链接操作;
- d) 远程操作;
- e) 远程操作服务元素;
- f) 调用者;
- g) 执行者;
- h) 联系类;
- i) 操作类。

3.4 CMIS 定义

本系列标准的这一部分使用在 GB/T 16644 中定义的下列术语:

- a) 属性;
- b) 公共管理信息服务元素;
- c) 公共管理信息服务;
- d) CMISE 服务提供者;
- e) CMISE 服务用户;
- f) 调用 CMISE 服务用户;
- g) 执行 CMISE 服务用户。

3.5 ACSE 定义

本系列标准的这一部分使用在 GB/T 16688 中定义的下列术语：

- a) 应用上下文；
- b) 应用联系；
- c) 联系。

3.6 表示定义

本系列标准的这一部分使用在 GB/T 15695 中定义的下列术语：

- a) 抽象语法；
- b) 传送语法。

4 符号和缩略语

ACSE	联系控制服务元素
APDU	应用协议数据单元
ASE	应用服务元素
ASN.1	抽象语法记法一
CMIP	公共管理信息协议
CMIPM	公共管理信息协议机
CMIS	公共管理信息服务
CMISE	公共管理信息服务元素
DCS	定义的上下文集
PCI	协议控制信息
PDU	协议数据单元
PICS	协议执行一致性声明
RO	远程操作
ROSE	远程操作服务元素
SMAE	系统管理应用实体

5 概述

公共管理信息协议(CIMP)规定了可以提供 GB/T 16644 中描述的操作和通知服务的协议元素,并且 GB/T 16644 还定义了公共管理信息服务(CMIS)。

5.1 提供的服务

在本系列标准的这部分中规定的协议支持在本系列标准中定义的服务,这些服务概括于表 1。

表 1 公共管理信息服务

服 务	类 型
M-CANCEL-GET	证实
M-EVENT-REPORT	证实/非证实
M-GET	证实
M-SET	证实/非证实
M-ACTION	证实/非证实
M-CREATE	证实
M-DELETE	证实

5.2 基础的服务

本系列标准的这一部分使用了在 ISO/IEC 9072-1:1989 中定义的远程操作服务元素(ROSE)的 RO-INVOKE、RO-RESULT、RO-ERROR 和 RO-REFLECT-U 服务。ROSE 假设使用在 GB EEEE 中定义的表示服务。证实的 CMIP 操作是应用所要求的操作类 2(异步)或操作类 1(同步)。操作类的选择是局部的事件。非证实的 CMIP 操作是操作类 5(异步,不报告结果)。CMIP 使用联系类 3。

如果协商扩充服务功能单元成功,则 RDSEapdus 可以被映射到表示服务而不是 P-DATA 服务。

注:例如,当 CMIP 操作被发送到对等 CMISE 服务用户时,可能需要修改表示定义的上下文集(DCS)。在这种情况下,运载 CMIP 操作的 ROSE APDU 就被映射到 P-ALTER-CONTEXT 服务,该服务也用于完成 DCS 的更改。

要求那些其他表示服务及如何使用他们的详情在联系上所用的应用上下文描述中加以描述。

5.2.1 设想 ACSE 的服务

本系列标准的这一部分设想使用联系控制服务元素的 A-ASSOCIATE、A-RELEASE、A-ABORT 和 A-P-ABORT 服务。

5.2.2 设想表示层服务

GB HHHH.2 设想使用表示层的 P-DATA 服务,以便传送 RO-INVOKE、RO-RESULT、RO-ERROR 和 RO-REJECT PDUS。

5.3 管理信息定义

本系列标准的这一部分定义了公共管理信息协议的抽象语法。协议运载的管理信息的定义不在本系列标准的这一部分中规定。

5.4 协议版本

本系列标准的这一部分定义了 CMIP 的第 2 版,第 2 版代替第 1 版。本系列标准的这一部分不定义第 2 版和第 1 版之间的任何互工作。

6 规程要素

本章提供了 CMIP 规程要素的定义。该规程定义了 CMIP PDU 的传送,该 CMIP PDU 的结构、编码和与 CMIS 服务原语的关系在第 7 章中规定。

公共管理信息协议机(CMIPM)接受 CMIS request 和 response 服务原语,并且发出在本章中规定的特定规程要素启动的 CMIP PUD。

CMIPM 应接受任何正确形成的 CMIP PDU,并借助 CMIS indication 和 confirmation 服务原语把它传送到执行的 CMISE 服务用户,以便处理。如果收到的 PDU 不是正确形成的或者不包括支持的通知或操作,那么返回 PDU,以表示收到 PDU 已被拒绝。

该规程只指出如何解释 CMIP PDU 中的各种字段,而不解释调用 CMISE 服务用户应该用请求的信息做什么,也不是解释执行 CMISE 服务用户应该如何处理调用。

6.1 联系建立

联系建立涉及两个 CMISE 服务用户,一个是联系启动者,而另一个是联系响应者。

除了其他内容外,CMISE 服务用户可以通过使用 GB/T 16688 的 A-ASSOCIATE 服务来启动联系建立。

应用上下文还规定了协调对应于不同的 ASE 的初始化信息所要求的规则。CMISE 的联系规则在附录 A 中规定。

6.2 远程操作

6.2.1 RO 规程要素

CMIP 规程要素依赖于下述基本的远程操作规程要素:

- a) 调用;
- b) 返回结果;
- c) 返回的差错;

- d) 用户拒绝;
- e) 提供者拒绝。

这些规程要素在 ISO/IEC 9072-2:1989 中进行全面描述。

表 2 规定了 CMIS 和 ROSE 参数之间的对应关系。

表 2 CMIS 和 ROSE 参数之间的对应关系

CMIS 参数	ROSE 参数
调用标识符	InvokeID
连接的标识符	Linked-ID

其他 CMIS 和 ROSE 参数之间的对应关系在第 7 章中规定。

6.2.2 RO 拒绝问题参数

RO 拒绝问题参数按如下映射或处理。

6.2.2.1 按如下所示,将 RO 拒绝用户、调用问题映射到 CMIS 差错代码。

表 3 将 RO 拒绝用户、调用问题映射到 CMISE 差错代码

RO-REJECT 参数	CMISE 错误代码
复制调用	duplicate invocation
类型的变量	mistyped argument
资源限制	resource limitation
未识别的操作	unrecognized operation

其他调用问题参数是一个本地事件。

6.2.2.2 其他 RO 拒绝参数将被当作局部事件来处理。

6.3 事件报告规程

6.3.1 调用

事件报告规程由 M-ENENT-REPORT request 原语来启动。

一旦接收到 M-ENENT-REPORT request 原语,CMIPM 应:

- a) 在证实方式下,构造请求 M-EventReport-Confirmed 操作的 APDU;否则构造一个请求 M-EVENT REPORT 操作的 APDU;
- b) 使用 RO 调用规程发送 APDU。

6.3.2 接收

一旦接收到请求 m-EventReport 或 m-EventReport-Confirmed 操作的 APDU,如果 APDU 已正确形成,则 CMIPM 应给 CMISE 用户发送 M-EVENT-REPORT indication 原语,用方式参数指明是否请求证实,否则 CMIPM 应构造包括差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.3.3 响应

在证实方式下,CMIPM 应接受一个 M-EVENT-REPORTER response 原语,并且应:

- a) 构造证实 M-EVENT-REPORT 通知的 APDU;
- b) 如果 M-EVENT-REPORT response 原语中的参数表明通知被接受,那么使用 RO-RESULT 规程来发送 APDU,否则使用 RO-ERROR 规程来发送 APDU。

6.3.4 收到响应

一旦接收到响应 M-EVENT-REPORT 通知的 APDU,如果 APDU 已正确形成,则 CMIPM 应发出 M-EVENT-REPORT confirmation 原语给 CMISE 服务用户,于是,完成这个通知规程,否则,CMIPM 应构造包含差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.4 获得规程

6.4.1 调用

获得规程由 M-GET request 原语来启动。

一旦接收到 M-GET request 原语,CMIPM 应:

- a) 构造请求 m-Get 操作的 APDU;
- b) 使用 RO-INVOKE 规程发送 APDU。

6.4.2 接收

一旦接收到请求 m-Get 操作的 APDU,如果 APDU 已正确形成,则 CMIPM 应发出 M-GET indication 原语给 CMISE 服务用户,否则,CMIPM 构造包含差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.4.3 响应

CMIPM 应:

- a) 接受零或多个包含链接标识符(linked-ID)的 M-GET response 原语,后面跟着不带 linked-ID 的单个 M-GET response 原语;
- b) 对于每一个包含链接标识符的 M-GET response 原语,CMIPM 应:
 - 构造申请 m-Linked-Reply 操作的 APDU,同时 LinkedReplyArgument 相应设置为 getListError、getResult 或 processingFailure;
 - 使用 RO-INVOKE 规程发送每一个 APDU。
- c) 对于不包括连接标识符的 M-GET request 原语,CMIPM 应:
 - 构造一个证实 m-Get 操作的 APDU;
 - 如果 M-GET response 原语中的参数表明这个操作已正确执行。则使用 RO-RESULT 规程来发送 APDU。如果 M-GET response 原语中的参数表明这个操作由于差错只成功地执行一部分,或者没有执行,则使用 RO-ERROR 规程来发送 APDU。

6.4.4 收到响应

一旦接收到响应 m-Get 操作的 APDU,CMIPM 应:

- a) 如果 APDU 包括 linked-ID 并且已正确形成,则发出 M-GET confirm 原语给 CMISE 服务用户;
- b) 如果 APDU 是最后一个响应(即不包括 linked-ID)并且已正确形成,则发出 M-GET confirm 原语给 CMISE 服务用户,从而完成了 M-GET 规程;
- c) 如果 APDU 没有正确形成,则构造包含差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.4.5 CancelGet 规程

6.4.5.1 调用

CancelGet 规程由 M-CANCEL-GET request 原语来启动

一旦接收到 M-CANCEL-GET request 原语,则 CMIPM 应:

- a) 构造请求 m-CancelGet 操作的 APDU;
- b) 使用 RO-INVOKE 规程来发送 APDU。

6.4.5.2 接收

一旦接收到请求 m-CancelGet 操作的 APDU,如果 APDU 已正确形成,则 CMIPM 应发出 M-CANCEL-GET indication 原语给 CMISE 服务用户,否则,CMIPM 构造包括差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.4.5.3 响应

CMIPM 应:

- a) 构造证实 m-CancelGet 操作的 APDU;

b) 如果 M-CANCEL-GET response 原语中的参数表明操作已经正确执行,则使用 RO-RESULT 规程来发送 APDU,否则使用 RO-ERROR 规程来发送 APDU。如果 M-CANCEL-GET 操作成功,则执行 CMISE 服务用户应停止发送对 M-GET 操作的链接应答,并且应发出包含“Operation cancelled”差错的 M-GET response 原语。

6.4.5.4 收到响应

一旦接收到一个响应 m-CancelGet 操作的 APDU,如果 APDU 已正确形成,则 CMIPM 应发出 M-CANCEL-GET confirm 原语给 CMISE 服务用户,否则,CMIP 构造包含差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.5 设置规程

6.5.1 调用

设置规程由 M-SET request 原语来启动。

一旦接收到 M-SET request 原语,CMIPM 应:

a) 在证实方式下,构造请求 m-Set-Confirmed 操作的 APDU,否则,构造请求 m-Set 操作的 APDU;

b) 使用 RO-INVOKE 规程来发送 APDU。

6.5.2 接收

一旦接收到请求 m-Set 或 m-Set-Confirmed 操作的 APDU,如果 APDU 已正确形成,CMIP 应发出 M-SET indication 原语给 CMISE 服务用户,同时方式参数指示是否请求证实,否则,构造包含差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.5.3 响应

在证实方式下,CMIPM 应:

a) 接受零或多个包含 linked-ID 的 M-SET response 原语,后面跟着不带 linked-ID 的单个 M-SET response 原语;

b) 对于每个包括 linked-ID 的 M-SET response 原语,CMIPM 应:

——构造请求 m-Linked-Reply 操作的 APDU,同时 LinkedReplyArgument 相应设置为 SetlistError、SetResult 或 processingFailure;

——使用 RO-INVOKE 规程来发送每个 APDU;

c) 对于不包括 linked-ID 的 M-SET response 原语,CMIPM 应:

——构造证实 m-set 操作的 APDU;

——如果 M-SET response 原语中的参数表明这个操作被正确执行,则使用 RO-RESULT 过程来发送 APDU。如果 M-SET response 原语中的参数表明这个操作由于差错只成功地执行了一部分,或者没有执行,则使用 RO-ERROR 规程来发送 APDU。

6.5.4 收到的响应

一旦接收到响应 m-Set-Confirmed 操作的 APDU,CMIPM 应:

a) 如果 APDU 包括 linked-ID,并已正确形成,则发出 M-SET confirm 原语给 CMISE 服务用户;

b) 如果 APDU 是最后一个响应(即不包括 linked-ID)并且已正确形成的话,则发出 M-SET confirmation 原语给 CMISE 服务用户,从而完成了 M-SET 规程;

c) 如果 APDU 未正确形成,则构造包含差错通知的 APDU,并使用 RO-REJECT-U 规程发送它。

6.6 动作规程

6.6.1 调用

动作规程由 M-ACTION request 原语来启动。

一旦接收到 M-ACTION request 原语,CMIPM 应:

a) 在证实方式下,构造请求 m-Action-Confirmed 操作的 APDU,否则,构造请求 m-Action 操作的

APDU;

- b) 使用 RO-INVOKE 规程来发送 APDU。

6.6.2 接收

一旦接收到请求 m-Action 或 m-Action-Confirmed 操作的 APDU,如果 APDU 已正确形成,CMIPM 应发出 M-ACTION indication 原语给 CMISE 服务用户,同时方式参数指示是否请求证实,否则,构造包含有差错通知的 APDU,并使用 RO-REJECT 规程来发送它。

6.6.3 响应

在证实方式下,CMIPM 应:

- a) 接受零或多个含有 linked-ID 的 M-ACTION response 原语,后面跟不带 linked-ID 的单个 M-ACTION response 原语;

- b) 对于每个含有 linked-ID 的 M-ACTION response 原语,CMIPM 应:

- 构造请求 m-Linked-Reply 操作的 APDU,同时 LinkedRepluArgument 相应设置为 action-Error,actionResult 或 processingFailure。
- 用 RO-INVOKE 规程发送每一个 APDU。

- c) 对于不包含 linked-ID 的 M-ACTION response 原语,CMIPM 应:

- 构造证实 m-Action 操作的 APDU。
- 如果 M-ACTION response 原语中的参数表明这个操作被正确执行,则使用 RO-ERROR 过程来发送 APDU。

6.6.4 收到响应

一旦接收到请求 m-Action-Confirmed 操作的 APDU,CMIPM 应:

- a) 如果 APDU 包括 linked-ID 并且已正确形成,则发出 M-ACTION confirm 原语给 CMISE 服务用户;

- b) 如果 APDU 是最后一个响应(即不包含 linked-ID)并已正确形成,发出 M-ACTION confirm 原语给 CMISE 服务用户,从而完成了 M-ACTION 规程;

- c) 如果 APDU 未正确形成,则构造含有差错通知的 APDU,并使用 RO-REJECT-U 规程发送它。

6.7 建立规程

6.7.1 调用

建立规程由 M-CREATE request 原语来启动。

一旦接收到 M-CREATE request 原语,CMIPM 应:

- a) 构造请求 m-Create 操作的 APDU;

- b) 使用 RO-INVOKE 规程来发送 APDU。

6.7.2 接收

一旦接收到请求 m-Create 操作的 APDU,如果 APDU 已正确形成,则 CMIPM 应发出 M-CREATE indication 原语给 CMISE 服务用户,否则,构造含有差错通知的 APDU,并使用 RO-REJECT-U 规程发送它。

6.7.3 响应

CMIPM 接受 M-CREATE response 原语并应:

- a) 构造证实 m-Create 操作的 APDU;

- b) 如果 M-CREATE response 原语中的参数表明这一操作被正确执行,则使用 RO-ERROR 规程来发送 APDU。

6.7.4 收到的响应

一旦接收到响应 m-Create 操作的 APDU,如果 APDU 已正确形成,CMIPM 应发出 M-CREATE confirm 原语给 CMISE 服务用户,从而完成了 M-CREATE 规程,否则,构造含有差错通知的 APDU,

并使用 RO-REJECT-U 规程发送它。

6.8 删除规程

6.8.1 调用

删除规程由 M-DELETE request 原语来启动。

一旦接收到 M-DELETE request 原语,CMIPM 应:

- a) 构造请求 m-Delete 操作的 APDU;
- b) 使用 RO-INVOKE 规程来发送 APDU。

6.8.2 接收

一旦接收到请求 m-Delete 操作的 APDU,如果 APDU 已正确形成,则 CMIPM 应发出 M-DELETE indication 给 CMISE 服务用户,否则,构造含有差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.8.3 响应

CMIPM 应:

- a) 接受零或多个包含 linked-ID 的 M-DELETE response 原语,后面跟着不带 linked-ID 的单个 M-DELETE response 原语;
- b) 对于每个含有 linked-ID 的 M-DELETE response 原语,CMIPM 应:
 - 构造请求 m-Linked-Reply 操作的 APDU,同时 LinkedReplyArgument 相应设置为 deleteError,deleteResult 或 processingFailure;
 - 使用 RO-INVOKE 规程来发送每个 APDU。
- c) 对于不包含 linked-ID 的 M-DELETE response 原语,CMIPM 应:
 - 构造证实 m-Delete 操作的 APDU;
 - 如果 M-DELETE response 原语中的参数表明这个操作被正确执行,则用 RO-RESULT 过程来发送 APDU,否则,使用 RO-ERROR 规程来发送 APDU。

6.8.4 收到的响应

一旦接收到响应 m-Delete 操作的 APDU,CMIPM 应:

- a) 如果 APDU 包含 linked-ID 并已正确形成,则发出 M-DELETE confirm 原语给 CMISE 服务用户;
- b) 如果 APDU 是最后一个响应(即不包括 linked-ID)并正确形成,则发出 M-DELETE confirm 原语给 CMISE 服务用户,从而完成了 M-DELETE 规程;
- c) 如果 APDU 未正确形成,则构造含有差错通知的 APDU,并使用 RO-REJECT-U 规程来发送它。

6.9 联系有序释放

任一 CMISE 服务用户可以使用 GB/T 16688 的 RELEASE 服务来启动联系的有序释放。

注:本规范与 BIND 操作的 ROSE 使用不同,在 Bind 操作中只有联系启动者才可使用 A-RELEASE 规程。

6.10 联系突然释放

任一 CMISE 服务用户可以使用 GB/T 16688 的 A-ABORT 服务来启动联系的突然释放。

CMISE 服务用户可以通过 GB/T 16688 的 A-ABORT 或 A-P-ABORT 服务接收到联系突然释放的指示。

7 抽象语法

本章规定 CMIP PDU 的抽象语法。

7.1 约定

抽象语法使用 GB/T 16262 规定的记法来定义。由本系列标准的这一部分使用或引用的 ASN.1

MACRO 产品并不运用无二义性的各方面措辞。

对于每个通过 CMIP PDU 传送的 CMISE 服务参数,都有一个与相应服务参数同名的 PDU 字段(一个 ASN.1 NamedType)(见 GB/T 16644),只有使用 ASN.1 所要求的差别,即字间的空格被去掉并且后随字的第一个字母加以大写,如“managed object class”就变成“mangedObjectClass”。为了使一些名字更短一些,某些字缩写如下:

ack=acknowledgement

arg=argument

id=identifier

info=information

sync=synchronization

7.2 CMISE 原语和 CMIP 操作之间的对应关系(见表 4)

表 4 CMISE 原语和 CMIP 操作之间的对应关系

CMIS 原语	方式	链接 ID	CMIP 操作
M-CANCEL-GET req/ind	证实	不适用	m-Cancel-Get-Confirmed
M-CANCEL-GET rsp/conf	不适用	不适用	m-Cancel-Get-Confirmed
M-EVENT-REPORT req/ind	非证实	不适用	m-EventReport
M-EVENT-REPORT req/ind	证实	不适用	m-EventReport-Confirmed
M-EVENT-REPORT rsp/conf	不适用	不适用	m-EventReport-Confirmed
M-GET req/ind	证实	不适用	m-Get
M-GET rsp/conf	不适用	不存在	m-Get
M-GET rsp/conf	不适用	存在	m-Linked-Reply
M-SET req/ind	非证实	不适用	m-Set
M-SET req/ind	证实	不适用	m-Set-Confirmed
M-SET rsp/conf	不适用	不存在	m-Set-Confirmed
M-SET rsp/conf	不适用	存在	m-Linked-Reply
M-ACTION req/ind	非证实	不适用	m-Action
M-ACTION req/ind	证实	不适用	m-Action-Confirmed
M-ACTION rsp/conf	不适用	不存在	m-Action-Confirmed
M-ACTION rsp/conf	不适用	存在	m-Linked-Reply
M-CREATE req/ind	证实	不适用	m-Create
M-CREATE rsp/conf	不适用	不适用	m-Create
M-DELETE req/ind	证实	不适用	m-Delete
M-DELETE rsp/conf	不适用	不存在	m-Delete
M-DELETE rsp/conf	不适用	存在	m-Linked-Reply

注:从 OPERATION 和 ERROR 宏到 ROSE 的映射在 ISO/IEC 9072:1989 中定义。

7.3 ACSE 用户数据

ACSE 协议(GB/T 16687 使用 ASN.1 来描述。“user information”使用 EXTERNAL 数据类型来定义。

7.3.1 A-ASSOCIATE 用户数据

传递到“user information”参数表示的 A-ASSOCIATE 的 CMIP 用户信息的编码定义如下:

CMIP-A-ASSOCIATE-Information {joint-iso-ccitt ms(9)cmip(1)modules(0)aAssociateUserInfo
(1)}

DEFINITIONS::=BEGIN

FunctionalUnits::=BIT STRING {multipleObjectSelection (0),
filter (1),
multipleReply (2),
extendedService (3),
cancelGet (4),

——当且仅当位 i 是 1,才支持的功能单元 i;

——在 A-ASSOCIATE 的用户信息参数中运载的信息。

CMIPUserInfo::=SEQUENCE {protocolVersion [0]IMPLICIT ProtocolVersion DEFAULT (version1),
functionalUnits [1]IMPLICIT FunctionalUnits DEFAULT {},
accessControl [2]EXTERNAL OPTIONAL,
userInfo [3]EXTERNAL OPTIONAL}

ProtocolVersion::=BIT STRING {version(0),
version(1)}

END

本系列标准的这一部分中不定义由 CMISE 服务用户供给的其他“用户信息”编码。

7.3.2 A-ABORT 用户数据

传递到“user information”参数表示的 A-ASSOCIATE 的 CMIP 用户信息的编码定义如下:

CMIP-A-ABORT-Information {joint-iso-ccitt ms(9)cmip(1)modules(0)aAbortUserInfo(2)}

DEFINITIONS::=BEGIN

——在 A-ABORT 的用户信息参数中运载的信息

CMIPAbortInfo::=SEQUENCE {abortSource [0]IMPLICIT CMIP AbortSource,
userInfo [1]EXTERNAL OPTIONAL}

CMIPAbortSource::=ENUMERATED {cmiseSeviceUser (0),
cmiseSeviceProvider (1)}

END

本系列标准的这一部分中不定义由 CMISE 服务用户供应的其他“用户信息”编码。

7.4 CMIP 数据单元

这个协议按照对等 CMISE 之间交换的公共管理信息协议数据单元来描述。PDU 使用 ASN.1 和在 ISO/IEC9072-1:1989 中定义的远程操作协议 OPERATION 和 ERROR 外部宏来规定。

——公共管理信息协议(CMIP)

CMIP-1 {joint-iso-ccitt ms(9)cmip(1)modules(0)protocol(3)}

DEFINITION::=BEGIN

——远程操作定义

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remote-Operations(4)notation(0)}

——远程操作服务定义

InvokeIDType FROM Remote-Operations-APDUs {joint-iso-ccitt-remote-Operations(4)apdus(1)}

——目录服务定义

DistinguishedName, RDNSequence FROM InformationFramework {joint-iso-ccitt ds(5)modules(1)informationFramework(1)};

——CMISE 操作

——在下面的操作中,自变量类型在相应的 ROSE APDU 中是强制的。

——Action 操作(M-ACTION)

m-Action OPERATION

ARGUMENT ActionArgument

::=localValue 6

m-Action-Confirmed OPERATION

ARGUMENT ActionArgument

RESULT ActionResult

——该结果是有条件的,关于条件见 GB/T 16644—1996 的 8.3.3.2.9

ERRORS { accessDenied, classInstanceConflict, complexityLimitation, invalidScope, invalidArgumentValue, invalidFilter, noSuchAction, noSuchArgument, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncnotSupported }

LINKED { m-linked-Reply }

::=localValue 7

m-CancelGet OPERATION

ARGUMENT

getInvokeId InvokeIDType

RESULT

ERRORS { mistypedOperation, noSuchInvokeId, processingFailure }

::=localValue(1)

——建立操作(M-CREATE)

m-Create OPERATION

ARGUMENT CreateArgument

RESULT CreateResult

——该结果是有条件的,关于条件见 GB/T 16644—1996 的 8.3.4.1.3

ERRORS { accessDenied, classInstanceConflict, duplicateManagedObjectInstance, invalidAttributeValue, invalidObjectInstance, missingAttributeValue, noSuchAttribute, noSuch

chObjectClass,
noSuchObjectInstance,noSuchReferenceObject,processingFailure}
::=localValue8

——删除操作(M-DELETE)

m-Delete OPERATION

ARGUMENT DeleteArgument
RESULT DeleteResult

——该结果是有条件的,关于条件见 GB/T 16644—1996 的 8.3.5.2.8

ERRORS{accessDenied,classInstanceConflict,complexityLimitation,invalidFilter,invalidScope,
noSuchObjectClass,noSuchObjectInstance,processingFailure,syncNotSupported}

LINKED{m-Linked-Reply}

::=localValue9

——事件报告操作(M-EVENT-REPORT)

m-EventReport OPERATION

ARGUMENT EventReportArgument

::=localValue0

m-EventReport-Confirmed OPERATION

ARGUMENT EventReportArgument

RESULT EventReportResult——可选

ERRORS{invalidArgumentValue,noSuchArgument,noSuchEventType,noSuchObjectClass,
noSuchObjectInstance,processingFailure}

::=localValue;1

——获得操作(M-GET)

m-Get OPERATION

ARGUMENT GetArgument

RESULT GetResult

——该结果是有条件的,关于条件见 GB/T 16644—1996 的 8.3.1.2.8

ERRORS{accessDenied,ClassInstanceConflict,complexityLimitation,getListError,invalidFilter,
invalidScope,noSuchObjectClass,noSuchObjectInstance,operationCancelled,process-
ingFailure,syncNotSupported}

LINKED{m-Linked-Reply}

::=localValue;3

——链接 M-GET,M-SET(Confirmed),M-ACTION(Confirmed)和 M-DELETE 的操作

m-Linked-Reply OPERATION

ARGUMENT LinkReplyArgument

::=localValue;2

——设置操作(M-SET)

m-Set OPERATION

ARGUMENT SetArgument
::=localValue:4

m-Set-Confirmed OPERATION

ARGUMENT SetArgument
RESULT SetResult

——该结果是有条件的,关于条件见 GB/T 16644—1996 的 8.3.2.2.9

ERRORS {accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope,
noSuchObjectClass, noSuchObjectInstanceConflict, processingFailure, setListError,
syncNotSupported}

LINKED {m-Linked-Reply}
::=localValue:5

——CMIS 差错定义

——在下面的差错中,除非另有指明,参数类型在相应的 ROSE APDU 中是强制的。

accessDenied ERROR

::=localValue:2

classInstanceConflict ERROR

PARAMETER BaseManagedObjectId
::=localValue:19

complexityLimitation ERROR

PARAMETER ComplexityLimitation——可选
::=localValue:20

duplicateManagedObjectInstance ERROR

PARAMETER ObjectInstance
::=localValue:11

getListError ERROR

PARAMETER GetlistError
::=localValue:7

invalidArgumentValue ERROR

PARAMETER InvalidArgument
::=localValue:15

invalidAttributeValue ERROR

PARAMETER Attribute
::=localValue:6

invalidFilter ERROR

PARAMETER CMISFilter

::=localValue;4

invalidObjectInstance ERROR

PARAMETER ObjectInstance

::=localValue;17

invalidScope ERROR

PARAMETER Scope

::=localValue;16

missingAttribute Value ERROR

PARAMETER SET OF AttributeId

::=localValue;18

mistypedOperation ERROR

::=localValue;21

noSuchAction ERROR

PARAMETER NoSuchAction

::=localValue;9

noSuchArgument ERROR

PARAMETER NoSuchArgument

::=localValue;14

noSuchAttribute ERROR

PARAMETER AttributeId

::=localValue;5

noSuchEventType ERROR

PARAMETER NoSuchEventtype

::=localValue;13

noSuchInvokedID ERROR

PARAMETER InvokeIDType

::=localValue;22

noSuchObjectClass ERROR

PARAMETER ObjectClass

::=localValue;0

```
noSuchReObjectInstance ERROR
    PARAMETER ObjectInstance
    ::= localValue;1
```

```
noSuchReferenceObject ERROR
    PARAMETER ObjectInstance
::=localValue;12
```

```
operationCancelled ERROR
::=localValue;23
```

```
processingFailure ERROR
    PARAMETER ProcessingFailure—— 任选
    ::=localValue;10
```

```
setListError ERROR
  PARAMETER setListError
  ::= localValue;8
```

```
syncNotSupported ERROR
    PARAMETER CMISSync
    ::=localValue:3
```

——支持类型定义

AccessControl ::= EXTERNAL

ActionArgument ::= SEQUENCE {	COMPONENTS OF	BaseManagedObjectID,
accessControl		[5]AccessControl OPTIONAL,
synchronization		[6]IMPLICIT CMISSync DEFAULT best-Effort,
scope		[7]Scope DEFAULT baseObject,
filter		CMISFilter DEFAULT and:{},
actionInfo		[12]IMPLICIT ActionInfo}

```

ActionError ::= SEQUENCE {
    managedObjectClass      ObjectClass OPTIONAL,
    managedObjectInstance   ObjectInstance OPTIONAL,
    currentTime              [5] IMPLICIT GeneralizedTime OPTIONAL,
    actionErrorInfo          [6] ActionErrorInfo }

```

$$\begin{aligned} \text{ActionErrorInfo} &::= \text{SEQUENCE}\{\text{errorStatus ENUMERATED}\{\text{accessDenied} & (2), \\ & \text{noSuchAction} & (9), \\ & \text{noSuchArgument} & (14), \end{aligned}$$

		invalidArgument	(15),}
errorInfo	CHOICE	{actionType	ActionTypeID,
		actionArgument	[0]NoSuchArgument,
		argumentValue	[1]InvalidArgumentValue}}

ActionInfo::=SEQUENCE{actionType ActionTypeID,
 actionInfoArg [4]ANY DEFINED BY actionType OPTIONAL}

ActionReply::=SEQUENCE{actionType ActionTypeID,
 actionReplyInfo [4]ANY DEFINED BY action Type}

ActionResult::=SEQUENCE{managedObjectClass ObjectClass OPTIONAL,
 managedObjectInstance ObjectInstance OPTIONAL,
 currentTime [5]IMPLICIT GeneralizedTime OPTIONAL,
 actionReply [6]IMPLICIT ActionReply OPTIONAL}

ActionTypeId::=CHOICE{globalForm [2]IMPLICIT OBJECT IDENTIFIER,
 localForm [3]IMPLICIT INTEGER}

——本系列标准的这一部分不为 localForm 分配任何值,在使用这种选择时

——允许的整数值及其含义应定义为使用它们的应用上下文的一部分。

Attribute::=SEQUENCE{attributeId AttributeId,
 attributeValue ANY DEFINED BY attributeID}

AttributeError::=SEQUENCE{errorStatus ENUMERATED{accessDenied (2),
 noSuchAttribute (5),
 invalidAttribute (6),
 invalidOperation (24),
 invalidOperator(25)},
 modifyOperator[2]IMPLICIT ModifyOperator OPTIONAL,

——对于 invalidOperator&invalidOperation 存在

attribute AttributeId,
attributeValue ANY DEFINED BY attributeId OPTIONAL
——对于 setToDefault 不存在
}

AttributeID::=CHOICE{globalForm [0]IMPLICIT OBJECT IDENTIFIER,
 localForm [1]IMPLICIT INTEGER}

——本系列标准的这一部分不为 localForm 分配任何值,在使用这种选择时

——允许的整数值及其含义应定义为使用它们的应用上下文的一部分。

AttributeIDError ::= SEQUENCE { errorStatus ENUMERATED { accessDenied (2),
noSuchAttribute (5) },
attributedID AttributeId }

BaseManagedObjectId ::= SEQUENCE { baseManageObjectClass ObjectClass,
BaseManageObjectInstance ObjectInstance }

CMISFilter ::= CHOICE { item [8] FilterItem,
and [9] IMPLICIT SET OF CMISFilter,
or [10] IMPLICIT SET OF CMISFilter,
not [11] CMISFilter }

CMISsync ::= ENUMERATED { bestEffort (0),
atomic (1) }

ComplexityLimitation ::= SET { scope [0] Scope OPTIONAL,
filter [1] CMISFilter OPTIONAL,
sync [2] CMISsync OPTIONAL }

CreateArgument ::= SEQUENCE {
managedObjectClass ObjectClass,
CHOICE { manageObjectInstance ObjectInstance,
superiorObjectInstance [8] ObjectInstance } OPTIONAL,
accessControl [5] AccessControl OPTIONAL,
referenceObjectInstance [6] ObjectInstance OPTIONAL,
attributeList [7] IMPLICIT SET OF Attribute OPTIONAL }

CreateResult ::= SEQUENCE { managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL,
——如果略去 CreateArgument, 则应
返回
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
attributeList [6] IMPLICIT SET OF Attribute OPTIONAL }

DeleteArgument ::= SEQUENCE { COMPONENTS OF BaseManageObjectId,
accessControl [5] AccessControl OPTIONAL,
synchronization [6] IMPLICIT CMISsync DEFAULT
bestEffort,

scope	[7]Scope DEFAULT baseObject,
filter	CMISFilter DEFAULT and: {}
DeleteError ::= SEQUENCE {	
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5]IMPLICIT GeneralizedTime OPTIONAL,
deleteErrorInfo	[6]ENUMERATED {accessDenied(2)}
DeleteResult ::= SEQUENCE {	
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5]IMPLICIT GeneralizedTime OPTIONAL}
EventReply ::= SEQUENCE {	
eventType	EventTypeId,
eventReplyInfo	[8]ANY DEFINED BY eventType OPTIONAL}
EventReportArgument ::= SEQUENCE {	
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
eventTime	[5]IMPLICIT GeneralizedTime OPTIONAL,
eventType	EventTypeId,
eventInfo	[8]ANY DEFINED BY eventType OPTIONAL}
EventReportResult ::= SEQUENCE {	
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5]IMPLICIT GeneralizedTime OPTIONAL,
eventReply	EventReply OPTIONAL}
EventTypeID ::= CHOICE {	
globalForm	[6]IMPLICIT OBJECT IDENTIFIER,
localForm	[7]IMPLICIT INTEGER}

——本系列标准的这一部分不为 localForm 分配任何值,在使用这种选择时

——允许的整数值及其含义应定义为使用它们的应用上下文的一部分。

FilterItem ::= CHOICE {

 equality [0]IMPLICIT Attribute,

 substrings [1]IMPLICIT SEQUENCE OF CHOICE {

 initialString [0]IMPLICIT SEQUENCE {

 attributId AttributId,

 string ANY DEFINED BY attributeId},

anyString [1]IMPLICIT SEQUENCE{
 attributeId AttributeId,
 string ANY DEFINED BY attributeId},
 finalString [2]IMPLICIT SEQUENCE{
 attributeId AttributeId,
 string ANY DEFINED BY attributeId}},
 greaterOrEqual [2]IMPLICIT Attribute,——确立值 $\geq$ 属性值
 lessOrEqual [3]IMPLICIT Attribute,——确立值 $\leq$ 属性值
 present [4]AttributeId,
 subsetOf [5]IMPLICIT Attribute,——确立值是属性值的子集
 supersetOf [6]IMPLICIT Attribute,——确立值是属性值上级的集合
 nonNullSetIntersection [7]IMPLICIT Attribute}

GetArgument::=SEQUENCE{COMPONENTS OF BaseManagedObjectId,
 accessControl [5]AccessControl OPTIONAL,
 synchronization [6]IMPLICIT CMISync DEFAULT bestEf-
 fort,
 scope [7]Scope DEFAULT baseObject,
 filter CMISFilter DEFAULT and:{},
 attributeIDList [12]IMPLICIT SET OF AttributeId OPTION-
 AL}

GetInfoStatus::=CHOICE{attributeIdError [0]IMPLICIT AttributeIdError,
 attribute [1]IMPLICIT Attribute}

GetListError::=SEQUENCE{managedObjectClass Objectclass OPTIONAL,
 managedObjectInstance ObjectInstance OPTIONAL,
 currentTime [5]IMPLICIT GeneralizedTime OPTIONAL,
 getInfoList [6]IMPLICIT SET OF GetInfoStatus}

GetResult::=SEQUENCE{managedObjectClass ObjectClass OPTIONAL,
 managedObjectInstance ObjectInstance OPTIONAL,
 currentTime [5]IMPLICIT GeneralizedTime OPTIONAL,
 attributeList [6]IMPLICIT SET OF Attribute OPTIONAL,

InvalidArgumentValue::=CHOICE{actionValue [0]IMPLICIT ActionInfo,
 eventValue [1]IMPLICIT SEQUENCE{
 eventType EventTypeId,
 eventInfo [8]ANY DEFINED BY
 eventType
 OPTIONAL}}

LinkedReplyArgument::=CHOICE{getResult [0]IMPLICIT GetResult,

getListError	[1]IMPLICIT GetListError,
setResult	[2]IMPLICIT SetResult,
setListError	[3]IMPLICIT SetListError,
actionResult	[4]IMPLICIT ActionResult,
processingFailure	[5]IMPLICIT ProcessingFailure,
deleteResult	[6]IMPLICIT DeleteResult,
actionError	[7]IMPLICIT ActionError,
deleteError	[8]IMPLICIT DeleteError}

ModifyOperator ::= INTEGER {
 replace (0),
 addValues (1),
 removeValue (2),
 setToDefault (3)}

NoSuchAction ::= SEQUENCE {
 managedObjectClass ObjectClass,
 actionType ActionTypeId}

NoSuchArgument ::= CHOICE {
 actionId [0]IMPLICIT SEQUENCE {
 managedObjectClass ObjectClass OPTIONAL,
 actionType ActionTypeId},
 eventId [1]IMPLICIT SEQUENCE {
 managedObjectClass ObjectClass OPTIONAL,
 eventType EventTypeId}}

NoSuchEventtype ::= SEQUENCE {
 managedObjectClass ObjectClass,
 eventType EventTypeId}

ObjectClass ::= CHOICE {
 globalForm [0]IMPLICIT OBJECT IDENTIFIER,
 localForm [1]IMPLICIT INTEGER}

- 本系列标准的这一部分不为 localForm 分配任何值,在使用这种选择时
- 允许的整数值及其含义应定义为使用它们的应用上下文的一部分。

ObjectInstance ::= CHOICE {
 distinguishedName [2]IMPLICIT DistinguishedName,
 nonSpecificForm [3]IMPLICIT OCTET STRING,
 localDistinguishedName [4]IMPLICIT RDNSequence}

- localDistinguishedName 是在开放系统之间的通信
- 上下文中无二义地标识被管理客体所需的可辨别名中的一部分。

ProcessingFailure ::= SEQUENCE {
 manageObjectClass ObjectClass,
 managedObjectInstance ObjectInstance OPTIONAL,
 specificErrorInfo [5]SpecificErrorInfo}

Scope::=CHOICE{INTEGER{baseObject (0),
firstLevelOnly (1),
wholeSubtree (2)},
individualLevels [1]IMPLICIT INTEGER,——POSITIVE 整数表示被选的
级别
baseToNthLevel [2]IMPLICIT INTEGER}——POSTIVE 整数 N 表示级别的
范围
——(0—N)被选择

——对 individualLevels 和 baseToNthLevel,0值与 baseObject 语义相同

——对 individualLevels,1值与 firstLevelOnly 语义相同

SetArgument::=SEQUENCE{COMPONENTS OF BaseManageObjectId,
accessControl [5]AccessControl OPTIONAL,
synchronization [6]IMPLICIT CMISync DEFAULT best-
Effort,
scope [7]Scope Default base Object,
filter CMISFilter DEFAULT and:{},
modificationList [12]IMPLICIT SET OF SEQUENCE{
modifyOperator [2]IMPLICIT ModifyOperator DEFAULT
replace,
attributeID AttributeId,
attributeValue ANY DEFINED BY attribute OPTIONAL
——对于 setToDe-
fault 缺少
}}

SetInfoStatus::=CHOICE{attribute [0]IMPLICIT AttributeError,
attribute [1]IMPLICIT Attribute}

SetListError::=SEQUENCE{managedObjectClass ObjectClass OPTIONAL,
manageObjectInstance ObjectInstance OPTIONAL,
currentTime [5]IMPLICIT GeneralizedTime OPTIONAL,
setInfoList [6]IMPLICIT SET OF Attribute OPTIONAL
AL}

SetResult::=SEQUENCE{managedObjectClass ObjectClass OPTIONAL,
manageObjectInstance ObjectInstance OPTIONAL,
currentTime [5]IMPLICIT GeneralizedTime OPTIONAL,
attributeList [6]IMPLICIT SET OF Attribute OPTIONAL}

SpecificErrorInfo::=SEQUENCE{errorId OBJECT IDENTIFIER,
errorInfo ANY DEFINED BY errorID}

END——CMIP 语法定义结束

7.5 CMIP 抽象语法定义

本系列标准的这一部分将 ASN.1 客体标识符值

{joint-iso-ms(9)cmip(1)cmip-pci(1)abstractSyntax(4)}

分配作为表示数据值集的抽象语法名,每个数据值或者是 ASN.1 类型的值

Remote-Operations-APDUs. ROSEapdus。

它按照 CMIP-1 的定义,使用所填充的变量成分在 ISO/IEC 9072-2:1989 中定义;或者是 ASN.1 类型中的某一个值,

——CMIP-A-ASSOCIATE-info, CMIPUserInfo;

——CMIP-A-ABORT 信息, CMIPabortInfo。

对应的 ASN.1 客体描述符值应为“CMIP-PCI”。

该抽象语法定义用来包括 X 定义的任何产品所分辨的所有数据类型,其中 X 是类型客体标识符。

ASN.1 客体标识符和客体描述符值

{joint-iso-ccitt asn1(1)basic-encoding(1)} 和“单个 ASN.1 类型的基本编码”具有该抽象语法时,

(在 GB KKKK 中分配给一个客体)能被用作作为传送语法名。

7.5.1 可扩充性规则

7.5.1.1 当处理进入的 CMIP-A-ASSOCIATE 信息时,接受 CMIPM 应:

——忽略所有在本部分中推荐的抽象语法中未定义的带标记的值,并;

——忽略在 BIT STRING 中所有不知道的位名赋值。

7.5.1.2 当表示数据值被修改并含有以下几个方面内容时可以使用抽象语法名。

——新系统管理操作;

——在 SET 或 SEQUENCE 中新的带标记的元素;

——在 BIT STRING 中新的位名赋值;

——新的命名整数;

——在 ENUMERATED 中新命名的枚举。

8 一致性

声称实现本标准规定的规程的系统应符合 8.1 和 8.2 中的要求。

8.1 静态要求

声称符合本系列标准这一部分的一种实现应伴随有与 GB/T 16644.2 规定的 PICS 形式表一致的 PICS。特别是,应指出该实现支持下列两种协议之一:

1) 支持该协议,要求它在启动者和执行者两者中都提供核心功能单元中定义的所有 CMIS 服务;
或

2) 支持该协议,要求它在启动者和/或执行者中提供核心功能单元中定义的某些 CMIS 服务;但并不是两个角色中的所有协议者相对于所有 CMISE 服务。

该实现应:

a) 为了声称一致性的角色中的这些服务,支持本系列标准的这一部分第 6 章中规定的规程要素;

b) 为了声称一致性的角色中的这些服务,支持与运送请求和/或响应所必需的协议数据单元相关的抽象语法;

c) 为了生成和解释抽象语法的“CMIP-PCI,”定义的 CMIP PDU,为了声称一致性的协议元素;支持由 GB KKKK 规定的编码规则导出的传送语法和命名的

{joint-iso-ccitt asn1(1)basic-encoding(1)},

- d) 支持 GB/T 16687 中定义的 ACSE 协议来建立和释放联系；
- e) 在任何包括 CMISE 和 ASE 的应用中，支持附录 A 中定义的规则；
- f) 支持 ISO/IEC 9072-2:1989 中定义的 ROSE 协议中的联系类 3；
- g) 如果多客体选择功能受到支持，则支持要求提供多应答功能单元的协议元素；
- h) 如果取消获得功能单元被支持，则支持提供 M-GET 服务所要求的协议元素。

8.2 动态要求

系统应：

- a) 为了声称一致性角色中的每个 CMIS 服务，支持规程要素；
- b) 当使用时，验证 CMIP PDU 中定义的任选安全参数；
- c) 当支持扩充服务功能单元时，按应用上下文的要求支持 GB/T 15696 中定义的表示协议；
- d) 当提出了范围时，支持规程要素用于多应答功能单元。

附录 A
(标准的附录)
用于 CMISE 的联系规则

A1 ACSE、会话和表示要求

A1.1 CMISE 要求在 GB/T 15695 中定义的核心功能单元。

A1.2 CMISE 要求在 GB/T 15128 中定义的核心会话功能单元和全双工会话功能单元。

A1.3 CMISE 要求在 GB/T 16688 和 GB/T 15695 中定义的 ACSE 正常方式和表示服务。

A2 联系初始化规则

A2.1 请求 Request

启动联系建立的 CMISE 服务用户应提供由 GB/T 16644 定义的 A-ASSOCIATE 用户信息。应使 CMIP 用户信息对 CMIPM 有效,CMIPM 应:

- a) 从所供给的信息中构造 CMIPUserInfo;
- b) 在 CMIPUserInfo 中通过设置对应于所支持的每版本的位来设置协议版本参数;
- c) 在 A-ASSOCIATE request 原语的用户信息参数中,将 CMIPUserInfo 作为一个独立 EXTERNAL;
- d) 在 A-ASSOCIATE confirm 原语中等待返回的 CMIS 特定用户信息。

A2.2 指示 Indication

一旦接收到 A-ASSOCIATE indication 原语,应使 CMIPUserInfo 参数对 CMIPM 有效,CMIPM 应:

- a) 检验至少能支持一个建议的协议版本;
- b) 验证任选的访问控制参数是否有效;
- c) 如果任何检验都失败,应通过将 A-ASSOCIATEResponse 原语中的失败参数设为“响应者拒绝(永久)”,而拒绝联系。从而未建立起联系 CMIPM 实例不再存在。
- d) 如果以上检验成功,下述信息(如果呈现在 CMIPUserInfo 中)应对 CMISE 服务用户有效:CMISE 服务提供者支持的功能单元、访问控制和用户信息。CMIPM 应等待来自 CMISE 服务用户的响应。

A2.3 响应 Response

A-ASSOCIATE response 原语指示“已接受”或“已拒绝”,如果已接受,它包括功能单元、访问控制和用户信息参数,应使之对 CMIPM 有效,CMIP 应:

- a) 构造响应所要求的 CMIPUserInfo。CMIPUserInfo 应包括指示所支持的所有 CMIP 版本的版本参数;
- b) 在 A-ASSOCIATE response 原语的用户信息参数中,将 CMIPUserInfo 作为一个独立 EXTERNAL;
- c) 如果联系响应指示“已接受”。双方同意的协议版本是由两个 CMIPM 支持的最高版本号的版本。CMIPM 应准备好接受 CMISE indication 原语;
- d) 如果联系响应指示“已拒绝”,CMIPM 实例应不再存在。

A2.4 证实 Confirmation

一旦接收到 A-ASSOCIATE confirmation 原语,应使 CMIPUserInfo 参数对 CMIPM 有效,CMIPM 应:

a) 如果联系证实指示成功,联系被建立,并且应使功能单元、访问控制和用户信息参数(如果呈现在证实中)对 association-initiator 有效。同意的功能单元符合于所指示的两个 CMISE 服务用户支持的功能单元,并且协议版本是两个 CMIPM 支持的最高版本号。

b) 如果联系证实指示失败,不建立联系并且 CMIPM 实例应不再存在。

A3 联系释放规则

任一 CMISE 服务用户可以启动联系释放。

A3.1 请求 Request

一旦接收到联系释放的请求,应使必要的 A-RELEASE 指示参数应对 CMIPM 有效,CMIPM 应停止接受服务请求并等待联系释放的证实。

A3.2 指示 Indication

一旦接收到 A-RELEASE indication 原语,应使必要的 A-RELEASE 指示参数应对响应 CMIPM 有效,响应 CMIPM 应等待联系释放响应。

A3.3 响应 Response

一旦接收到来自相应 CMISE 服务用户的联系释放响应,应使必要的 A-RELEASE 响应参数应对响应 CMIPM 有效。其后,CMIPM 实例应不再存在。

A3.4 证实 Confirmation

一旦接收到 A-RELEASE confirm 原语,应使必要的 A-RELEASE 证实参数应对启动的 CMIPM 有效,其后,CMIPM 实例应不再存在。

A4 联系放弃规则

任一 CMISE 服务用户可以启动联系突然终止。

在局部信息的基础上,如果基础服务通过 A-ABORT 运送无限用户信息的能力不存在,CMIPAbortInfo 参数不应包括在 A-ABORT 服务原语中。

A4.1 A-ABORT request

一旦接收到放弃联系的请求,应使包括 GB/T 16644 中定义的 A-ABORT 用户信息在内的必要的 A-ABORT 请求参数,对 CMIPM 有效,CMIPM 应:

- 从所提供的信息中构造 CMIPAbortInfo;
- 将 CMISEUserInfo 内的放弃源参数设置为 CMISE 服务用户;
- 在 A-ABORT request 原语的用户信息参数中,将 CMIPAbortInfo 作为一个独立的字段;
- 其后,CMIPM 实例应不再存在。

A4.2 A-ABORT Indication

一旦接收到 A-ABORT indication 原语,应使包括 CMIPAbortInfo 在内的必要的 A-ABORT indication 参数对 CMIPM 有效。其后,CMIPM 实例应不再存在。

A4.3 A-P-ABORT Indiation

一旦接收到 A-P-ABORT indication 原语,应使包括 GB/T 16644 中定义的 A-ABORT 用户信息在内的必要的 A-P-ABORT indication 参数对 CMIPM 有效。其后,CMIPM 实例应不再存在。

A4.4 CMIP 协议差错

在检测协议差错时,CMIPM 应:

- 构造 CMIPAbortInfo,同时将放弃源参数设置为 CMISE 服务提供者;
- 为 CMIP 服务用户指示已发生的协议差错;
- 在 A-ABORT request 原语的用户信息参数中将 CMIPAbortInfo 作为一个独立的字段;
- 其后,CMIPM 实例应不再存在。

附 录 B
(提示的附录)
扩充 ASN.1 语法

本附录描述如何将 ISO/IEC 9072-1:1989 的 OPERATION 和 ERROR 宏扩充为 ASN.1 数据类型和子类型。

如果在这些定义与第7章中的定义之间存在任何不一致,则第7章的定义优先。

——公共管理信息协议(CMIP)

CMIP-1 {joint-iso-ccitt ms(9)cmip(1)modules(0)protocol(3)}

DEFINITIONS ::= BEGIN

——远程操作定义

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint iso-ccitt remote-Operations(4)notation(0)}

——目录服务定义

DistinguishedName, RDNSequence FROM InformationFramework {joint-iso ccitt ds(5)modules(1)informationFramework(1);

——CMISE 操作

DOSEapdu ::= CHOICE {
 roiv-apdu [1] IMPLICIT ROIVapdu,
 rors-apdu [2] IMPLICIT RORSapdu,
 roer-apdu [3] IMPLICIT ROERapdu,
 rorj-apdu [4] IMPLICIT RORJapdu }

ROIVapdu ::= SEQUENCE {
 invokeID InvokeIDType,
 linked-ID [0] IMPLICIT InvokeIDType OPTIONAL,
 operation-value OPERATION,
 argument ANY DEFINED BY operation-value OPTIONAL }

RORSapdu ::= SEQUENCE {
 invokeID InvokeIDType,
 SEQUENCE {
 operation-value OPERATION,
 result ANY DEFINED BY operation-value } OPTIONAL }

ROERapdu ::= SEQUENCE {
 invokeID InvokeIDType,
 error-value ERROR,
 parameter ANY DEFINED BY error-value OPTIONAL }

RORJapdu ::= SEQUENCE { invokeID CHOICE { InvokeIDType, NULL },
 problem CHOICE { [0]IMPLICIT GeneralProblem, [1]IMPLICIT InvokeProblem, [2]IMPLICIT ReturnResultProblem, [3]IMPLICIT ReturnErrorProblem } }

InvokeIDType ::= INTEGER

——GeneralProblem、ReturnResultProblem 和 ReturnErrorProblem 代码的使用是本地问题

GeneralProblem ::= INTEGER { unrecognisedAPDU (0), —— ROSE 提供者被发现
 mistypedAPDU (1),
 badlyStructuredAPDU (2) }

InvokeProblem ::= INTEGER { duplicateInvocation (0), —— ROSE 用户被发现
 unrecognisedOperation (1),
 mistypedArgument (2),
 resourceLimitation (3),
 initiatorReleasing (4),
 unrecognisedLinkedID (5),
 linkedResponseUnexpected (6),
 unexpectedChildOperation (7) }

ReturnResultProblem ::= INTEGER { unrecognisedInvocation (0), —— ROSE 用户被发现
 resultResponseUnexpected (1),
 mistypeResult (2) }

ReturnErrorProblem ::= INTEGER { unrecognisedInvocation (0), —— ROSE 用户被发现
 errorResponseUnexpected (1),
 unrecognisedError (2),
 unexpectedError (3),
 mistypedParameter (4) }

——ASN.1规范的这部分提供 CMIP 所使用的 InvokeProblem 子类型的定义

InvokeProblem-CMIPUser ::= InvokeProblem (duplicateInvocation |
 unrecognisedOperation |
 mistypedArgument |
 resourceLimitation)

——ASN.1规范的这部分提供 CMIP 所使用的 ROIVapdu 和 RORSapdu 子类型的定义

——ROIVapdu 的子类型为所有 CMIP 通知和操作定义了操作值及该操作值所定义的自变量的允许值。

——RORSapdu 的子类型为所有 CMIP 通知和操作定义了操作值及该操作值所定义的结果的允许值。

m-Action OPERATION::=localValue:6

ROIV-m-Action::=ROIVapdu(WITH COMPONENTS

{inoveID	PRESENT,
linked-ID	ABSENT,
operation-value	(m-Action),
argument	(INCLUDES ActionArgument))}

m-Action-Confirmed OPERATION::=localValue:7

ROIV-m-Action-Confirmed::=ROIVapdu(WITH COMPONENTS

{inoveID	PRESENT,
linked-ID	ABSENT,
operation-value	(m-Action-Confirmed),
argument	(INCLUDES ActionArgument))}

RORS-m-Action-Confirmed::=RORSapdu(WITH COMPONENTS

{...	
invokeID	PRESENT,
—— 结果序列 ——	(WITH COMPONENTS
{operation-value	(m-Action-Confirmed),
result	(INCLUDES ActionResult))}
—— 仅当对 ROIV-m-Action-Confirmed ROIVapdu 只有单个	
—— 回答并且数据在 RORSapdu 中返回时才要求	
)}	

m-Cancel-Get OPERATION::=localValue:10

ROIV-m-Cancel-Get::=ROIVapdu(WITH COMPONENTS

{invokeID	PRESENT,
linked-ID	ABSENT,
operation-value	(m-Cancel-Get),
argument	(INCLUDES InvokeIDType))}

RORS-m-Cancel-Get::=RORSapdu(WITHCOMPONENTS

{invokeID	PRESENT
—— 对于 RORS-m-Cancel-Get 没有结果序列	
)}	

m-Create OPERATION::=localValue:8

ROIV-m-Create::=ROIVapdu(WITH COMPONENTS
 {invokeID PRESENT,
 linked-ID ABSENT,
 operation-value (m-Create),
 argument (INCLUDE CreateArgument)})

RORS-m-Create:=RORSapdu(WITH COMPONENTS
 {...,
 invokeID PRESENT,
 ——结果序列——(WITH COMPONENTS
 {operation-value(m-Create),
 result (INCLUDE CreateResult)})
 })

m-Delete OPERATION::=localValue:9

ROIV-m-Delete::=ROIVapdu(WITH COMPONENTS
 {invokeID PRESENT,
 linked-ID ABSENT,
 operation-value (m-Delete),
 argument (INCLUDES DeleteArgument)})

RORS-m-Delete::=RORSapdu(WITH COMPONENTS
 {...,
 invokeID PRESENT,
 ——结果序列——(WITH COMPONENTS
 {operation-value (m-Delete),
 result (INCLUDES DeleteResult)})
 ——仅当对 ROIV-m-DeleteROIVapdu 只有单个
 ——回答并且数据在 RORSapdu 中返回时才要求
 })

m-EventReport OPERATION::=localValue:0

ROIV-m-EventReport::=ROIVapde(WITH COMPONENTS
 {invokeID PRESENT,
 linked-ID ABSENT,
 operation-value (m-EventReport),
 argument (INCLUDES EventReportArgument)})

m-EvnetReport-Confirmed OPERATION::=localValue:1

ROIV-m-EventReport-Confirmed::=ROIVapde(WITH COMPONENTS

```

{invokeID      PRESENT,
linked-ID      ABSENT,
operation-value (m-EventReport-Confirmed),
argument       (INCLUDES EventReportArgument)))

```

RORS-m-EventReport-Confirmed::=RORSapdu(WITH COMPONENTS

```

{...,
  invokeID      PRESENT,
  ——结果序列—— (WITH COMPONENTS
    {operation-value (m-EvnetReport-Confirmed),
    result          (INCLUDES EventReportResult)})
  ——仅当数据在 RORSapdu 中返回时才要求
  })

```

m-Get OPERATION::=localValue;3

ROIV-m-Get::=ROIVapdu(WITH COMPONENTS

```

{invokeID      PRESENT,
linked-ID      ABSENT,
operation-value (m-Get),
argument       (INCLUDES GetArgument)})

```

RORS-m-Get::=RORSapdu(WITH COMPONENTS

```

{...,
  invokeID      PRESENT,
  ——结果序列—— (WITH COMPONENTS
    {operation-value (m-Get),
    result          (INCLUDES GetResult)})
  ——仅当有单个回答给 ROIV-m-Get ROIVapdu 时才要求
  })

```

m-Linked-Reply OPERATION::=localValue;2

ROIV-m-Linked-Reply::=ROIVapdu(WITH COMPONENTS

```

{invokeID      PRESENT,
linked-ID      ABSENT,
operation-value (m-Linked-Reply),
argument       (INCLUDE LinkedReplyArgument)})

```

——ASN.1规范的这一部分提供 CMIP 所使用的 ROIV-m-Linked-Reply 子类型的定义

——ROIV-m-Linked-Reply ROIVapdu 的子类型为特定 CMIP 链接回答操作

——定义了操作值所定义的自变量的允许值

ROIV-m-Linked-Reply-Action::=ROIV-m-Linked-Reply(WITH COMPONENTS

{invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument(WITH COMPONENTS
	{getResult ABSENT,
	getListError ABSENT,
	setResult ABSENT,
	setListError ABSENT,
	actionResult PRESENT,
	processingFailure PRESENT,
	deleteResult ABSENT,
	actionError PRESENT,
	deleteError ABSENT}))
	)))

ROIV-m-Linked-Reply-Delete::=ROIV-m-Linked-Reply(WITH COMPONENTS

{invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument(WITH COMPONENTS
	{getResult ABSENT,
	getListError ABSENT,
	setResult ABSENT,
	setListError ABSENT,
	actionResult ABSENT,
	processingFailure PRESENT,
	deleteResult PRESENT,
	actionError ABSENT,
	deleteError PRESENT}))
	)))

ROIV-m-Linked-Reply-Get::=ROIV-m-Linked-Reply(WITH COMPONENTS

{invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument(WITH COMPONENTS
	{getResult PRESENT,
	getListError PRESENT,

```

        setResult      ABSENT,
        setListError   ABSENT,
        actionResult   ABSENT,
        processingFailure ABSENT,
        deleteResult    PRESENT,
        actionError     ABSENT,
        deleteError     ABSENT}))
    )))

```

ROIV-m-Linked-Reply-Set::=ROIV-m-Linked-Reply(WITH COMPONENTS

```

    {invokeID      PRESENT,
    linked-ID      PRESENT,
    operation-value (m-Linked-Reply),
    argument       (INCLUDES LinkedReplyArgument(WITH COMPONENTS
        {getResult      ABSENT,
        getListError    ABSENT,
        setResult       PRESENT,
        setListError    PRESENT,
        actionResult    ABSENT,
        processingFailure PRESENT,
        deleteResult     ABSENT,
        actionError     ABSENT,
        deleteError     ABSENT}))
    )))

```

m-Set OPERATION::=localValue:4

ROIV-m-Set::=ROIVapdu(WITH COMPONENTS

```

    {invokeID      PRESENT,
    linked-ID      ABSENT,
    operation-value (m-Set),
    argument       (INCLUDES SetArgument))

```

m-Set-Confirmed OPERATION::=localValue:5

ROIV-m-Set-Confirmed::=ROIVapdu(WITH COMPONENTS

```

    {invokeID      PRESENT,
    linked-ID      ABSENT,
    operation-value (m-Set-Confirmed),
    argument       (INCLUDES SetArgument))

```

RORS-m-Set-Confirmed::=RORSapdu(WITH COMPONENTS

```
{...,
  invokeID      PRESNET,
  ——结果序列—— (WITH COMPONENTS
    {operation-value(m-Set-Confirmed),
      result      (INCLUDES SetResult)})
  ——仅当对 ROIV-m-Set-Confirmed ROIVapdu 只有单个
  ——回答并且数据在 RORSapdu 中返回时才要求
  })
```

——ASN.1规范的这部分提供 CMIP 所使用的 ROERapdu 子类型的定义

——ROERapdu 的子类型为所有 CMIP 通知和操作定义了差错值及该差错值所定义的参数的允许值。

accessDenied ERROR::=localValue:2

ROER-accessDenied::=ROERapdu(WITH COMPONENTS

```
{invokeID      PRESENT,
  error-value    (accessDenied)})
```

——ROERapdu 只在响应 ROIV-m-Get、ROIV-m-Set-Confirmed、

——ROIV-m-Action-Confirmed、ROIV-m-Create 和 ROIV-m-DeleteROIVapdus 时才被返回

classInstanceConflict ERROR::=localValue:19

ROER-classInstanceConflict::=ROERapdu(WITH COMPONENTS

```
{invokeID      PRESENT,
  error-value    (ClassInstanceConflict),
  parameter      (INCLUDES BaseManagedObjectID)})
```

——ROERapdu 只在响应 ROIV-m-Get、ROIV-m-Set-Confirmed、

——ROIV-m-Action-Confirmed、ROIV-m-Create 和 ROIV-m-Delete ROIVapdus 时才被返回

complexityLimitation ERROR::=localValue:20

ROER-complexityLimitation::=ROERapdu(WITH COMPONENTS

```
{invokeID      PRESENT,
  error-value    (complexityLimitation),
  parameter      (INCLUDES ComplexityLimitaion)OPTIONAL}}
```

——ROERapdu 只在响应 ROIV-m-Get、ROIV-m-Set-Confirmed、

——ROIV-m-Action-Confirmed、ROIV-m-Create 和 ROIV-m-Delete ROIVapdus 时才被返回

duplicateManagedObjectInstance ERROR::=localValue:11

ROER-duplicateManagedObjectInstance::=ROERapdu(WITH COMPONENTS


```

    {invokeID    PRESENT,
      error-value (duplicateManagedObjectInstance),
      parameter   (INCLUDES ObjectInstance))}

```

——ROERapdu 只在响应 ROIV-m-Create ROIVapdu 时才被返回

getListError ERROR::=localValue:7

```

ROER-getListError::=ROERapdu(WITH COMPONENTS
    {invokeID    PRESENT,
      error-value (getListError),
      parameter   (INCLUDES GetListError)})

```

——ROERapdu 只在响应 ROIV-m-Get ROIVapdu 时才被返回

invalidArgumentValue ERROR::=localValue:15

```

ROER-invalidArgumentValue::=ROERapdu(WITH COMPONENTS
    {invokeID    PRESENT,
      error-value (invalidArgumentValue),
      parameter   (INCLUDES InvalidArgumentValue)})

```

——ROERapdu 只在响应 ROIV-m-EventReport-Confirmet

——和 ROIVapdu-m-Action-Confirmed ROIVapdus 时才被返回

invalidAttributeValue ERROR::=localValue:6

```

ROER-invalidAttributeValue::=ROERapdu(WITH COMPONENTS
    {invokeID    PRESENT,
      error-value (invalidAttributeValue),
      parameter   (INCLUDES Attribute)})

```

——ROERapdu 只在响应 ROIV-m-Create ROIVapdu 时才被返回

invalidFilter ERROR::=localValue:4

```

ROER-invalidFilter::=ROERapdu(WITH COMPONENTS
    {invokeID    PRESENT,
      error-value (invalidFilter),
      parameter   (INCLUDES CMISFilter)})

```

——ROERapdu 只在响应 ROIV-m-Get、ROIV-m-Set-Confirmed、

——ROIV-m-Action-Confirmed 和 ROIV-m-Delete ROIVapdus 时才被返回

invalidObjectInstance ERROR::=localValue:17

```

ROER-invalidObjectInstance::=ROERapdu(WITH COMPONENTS
    {invokeID    PRESENT,

```

error-value (invalidObjectInstance),
parameter (INCLUDES ObjectInstance))

——ROERapdu 只在响应 ROIV-m-Create ROIVapdus 时才被返回

invalidScope ERROR::=localValue;16

ROER-invalidScope::=ROERapdu(WITH COMPONENTS
{invokeID PRESENT,
error-value (invalidScope),
parameter (INCLUDES Scope)})

——ROERapdu 只在响应 ROIV-m-Get、ROIV-m-Set-Confirmed、

——ROIV-m-Action-Confirmed 和 ROIV-m-Delete ROIVapdus 时才被返回

missingAttribute ERROR::=localValue;18

ROER-missingAttributeValue::=ROERapdu(WITH COMPONENTS
{invokeID PRESENT,
error-value (missingAttributeValue),
parameter (INCLUDES SET OF AttributeId)})

——ROERapdu 只在响应 ROIV-m-Create ROIVapdus 时才被返回

mistypedOperation ERROR::=localValue;21

ROER-mistypedOperation::=ROERapdu(WITH COMPONENTS
{invokeID PRESENT,
error-value (mistypedOperation)})

——ROERapdu 只在响应 ROIV-m-Cancel-Get ROIVapdus 时才被返回

noSuchAction ERROR::=localValue;9

ROER-noSuchAction::=ROERapdu(WITH COMPONENTS
{invokeID PRESENT,
error-value (noSuchAction),
parameter (INCLUDES NoSuchAction)})

——ROERapdu 只在响应 ROIV-m-Action-Confirmed ROIVapdus 时才被返回

noSuchArgument ERROR::=localValue;14

ROER-noSuchArgument::=ROERapdu(WITH COMPONENTS
{invokeID PRESENT,
error-value (noSuchArgument),
parameter (INCLUDES NoSuchArgument)})

——ROERapdu 只在响应 ROIV-m-EventReprot-Confirmed

——和 ROIV-m-Action-Confirmed ROIVapdus 时才被返回

noSuchAttribute ERROR::=localValue;5

ROER-noSuchAttribute::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (noSuchAttribute),
 parameter (INCLUDES AttributeId)})

——ROERapdu 只在响应 ROIV-m-Create ROIVapdus 时才被返回

noSuchEventType ERROR::=localValue;13

ROER-noSuchEventType::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (noSuchEventType),
 parameter (INCLUDES NoSuchEventType)})

——ROERapdu 只在响应 ROIV-m-EventReport-Confirmed ROIVapdus 时才被返回

noSuchInvokeId ERROR::=localValue;22

ROER-noSuchInvokeId::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (noSuchInvokeId),
 parameter (INCLUDES InvokeIDType)})

——ROERapdu 只在响应 ROIV-m-Cancel-Get ROIVapdus 时才被返回

noSuchObjectClass ERROR::=localValue;0

ROER-noSuchObjectClass::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (noSuchObjectClass),
 parameter (INCLUDES ObjectClass)})

——ROERapdu 只在响应 ROIV-m-EventReport-Confirmed、ROIV-m-Get

—— ROIV-m-Set-Confirmed、ROIV-m-Action-Confirmed、ROIV-m-Create、ROIV-m-DELETE
 ROIVapdus 时才被返回

noSuchObjectInstance ERROR::=localValue;1

ROER-noSuchObjectInstance::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (noSuchObjectInstance),
 parameter (INCLUDES ObjectInstance)})

——ROERapdu 只在响应 ROIV-m-EventReport-Confirmed、ROIV-m-Get

—— ROIV-m-Set-Confirmed、ROIV-m-Action-Confirmed、ROIV-m-Create、ROIV-m-DELETE ROIVapdus 时才被返回

noSuchReferenceObject ERROR::=localValue:12

ROER-noSuchReferenceObject::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (noSuchReferenceObject),
 parameter (INCLUDES ObjectInstance)})

——ROERapdu 只在响应 ROIV-m-Create ROIVapdus 时才被返回

operationCancelled ERROR::=localValue:23

ROER-operationCancelled::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (operationCancelled)})

——ROERapdu 只在响应 ROIV-m-Get ROIVapdus 时才被返回

processingFailure ERROR::=localValue:10

ROER-processingFailure::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (processingFailure),
 parameter (INCLUDES ProcessingFailure)OPTINAL)})

——ROERapdu 只在响应 ROIV-m-EventReport-Confirmed、ROIV-m-Get

—— ROIV-m-Set-Confirmed、ROIV-m-Action-Confirmed、ROIV-m-Create、ROIV-m-DELETE ROIVapdus 时才被返回

setListError ERROR::=localValue:8

ROER-setListError::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (setListError),
 parameter (INCLUDES SetListError)})

——ROERapdu 只在响应 ROIV-m-Set-Confirmed ROIVapdus 时才被返回

syncNotSupported ERROR::=localValue:3

ROER-syncNotSupported::=ROERapdu(WITH COMPONENTS
 {invokeID PRESENT,
 error-value (syncNotSupported),
 parameter (INCLUDES CMISSync)})

——ROERapdu 只在响应 ROIV-m-EventReport-Confirmed、ROIV-m-Get

—— ROIV-m-Set-Confirmed、ROIV-m-Action-Confirmed、ROIV-m-Create、ROIV-m-Delete ROIVap-
dus

—— 时才能被返回

—— 为了完成在本附录中提供的抽象语法规范,通过引用把7.4中支持类型的定义包括进去

END——CMIP 语法定义

附 录 C

(提示的附录)

CMISE ROSE APDU 例子

本附录提供运载 CMIP 信息的一些 ROSE APDU 的完整扩充例子。

这些例子为本系列标准这一部分的用户提供指南。

—— 用于 CMISE 证实动作操作的 ROIVapdu

```
ROIVapdu-example ::= [1]IMPLICIT SEQUENCE{
    invokeID          InvokeIDType,
    operation-value    INTEGER {m-Action-Confirmed(7)},
    argument           SEQUENCE{
        COMPONENTS OF BaseManagedObjectId,
        accessControl  [5]AccessControl OPTIONAL,
        synchronization [6]IMPLICIT CMISync OPTOINAL,
        scope          [7]Scope DEFAULT baseObject,
        filter          CMISFilter DEFAULT and:{},
        actionInfo      [12]IMPLICIT SEQUENCE{
            actionType  ActionTypeId,
            actionInfoArg [4]ANY DEFINED BY actionType OPTIONAL}}}
```

—— 用于 CMISE 证实动作操作的 RORSapdu

```
RORSapdu-example ::= [2]IMPLICIT SEQUENCE{
    invokeID          InvokeIDType,
    SEQUENCE{
        operation-value    INTEGER {m-Action-Confirmed(7)},
        result             SEQUENCE{
            managedObjectClass  ObjectClass OPTIONAL,
            managedObjectInstance ObjectInstance OPTIONAL,
            currentTime         [5]IMPLICIT GeneralizedTime OPTIONAL,
            actionReply         [6]IMPLICIT SEQUENCE{
                actionType      ActionTypeId,
                actionReplyInfo [4]ANY DEFINED BY actionType OPTIONAL}
            OPTIONAL}
```

——用于 CMISE 链接回答为了证实动作操作的 ROIVapdu

```
ROIVapdu-linked-example ::= [1]IMPLICIT SEQUENCE{  
    invokeID      InvokeIDType,  
    linked-ID     [0]IMPLICIT InvokeIDType,  
    operation-value INTEGER {m-Action-Confirmed(7)},  
    argument      CHOICE{  
        actionResult      [4]IMPLICIT ActionResult,  
        processingFailure  [5]IMPLICIT ProcessingFailure,  
        actionError       [7]IMPLICIT ActionError}}
```

——当出现 noSuchAction 差错时,用于 CMISE 证实动作操作的 ROIVapdu

```
ROERapdu-example ::= [3]IMPLICIT SEQUENCE{  
    invokeID      InvokeIDType,  
    error-value   INTEGER {noSuchAction(9)},  
    parameter     SEQUENCE {managedObjectClass ObjectClass OPTIONAL,  
                           actionId      ActionTypeId}},
```

中 华 人 民 共 和 国
国 家 标 准
信息技术 开放系统互连
公共管理信息协议 第1部分:规范
GB/T 16645.1—1996

*

中国标准出版社出版
北京复兴门外三里河北街16号
邮政编码:100045
电 话:68522112

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

*

开本 880×1230 1/16 印张 3 字数 68千字
1997年8月第一版 1997年8月第一次印刷
印数 1—1 000

*

书号: 155066·1-13970 定价 20.00 元

*

标 目 313—33



GB/T 16645.1—1996