



# 中华人民共和国通信行业标准

YD/T 2421-2012

---

## 域名注册协议主机供应技术要求

Technical requirements for host mapping protocol in domain name  
registration protocol

2012-12-28 发布

2013-03-01 实施

---

中华人民共和国工业和信息化部 发布

目 次

前 言.....II

引 言.....III

1 范围.....1

2 术语、定义和缩略语.....1

2.1 术语和定义.....1

2.2 缩略语.....2

3 协议概述.....2

4 协议要求.....2

4.1 域名对象和主机对象的关系.....2

4.2 对象属性.....2

4.3 EPP 命令映射.....4

4.4 形式语法.....7

4.5 中文化规定.....8

附录 A（资料性附录） 例子.....9

参考文献.....23

## 前 言

随着互联网的不断发展，域名系统越来越成为影响互联网稳定性的基础设施和核心系统。而域名注册服务为域名系统提供数据源并进行域名管理，对整个域名系统的稳定和安全有着至关重要的作用。因此，十分有必要为域名注册服务中的各技术方面制定相应的标准规范，以便为域名注册系统的安全稳定运行提供可靠的参考和保证。

本标准按照 GB/T 1.1-2009 给出的规则起草。

本标准是“域名注册”系列标准之一，该系列标准的名称和结构如下：

1. 《域名注册总体技术要求》
2. 《域名注册审核要求》
3. 《域名注册数据存储技术要求》
4. 《域名注册系统服务水平要求》
5. 《域名注册服务系统安全技术要求》
6. 《域名注册系统安全防护要求》
7. 《域名注册系统安全防护检测要求》
8. 《域名注册协议主机供应技术要求》
9. 《域名注册协议基于TCP的传输技术要求》
10. 《域名注册协议可扩展供应协议技术要求》
11. 《域名注册协议域名供应技术要求》
12. 《域名注册协议联系人供应技术要求》

本标准由中国通信标准化协会提出并归口。

标准起草单位：中国互联网络信息中心（CNNIC）。

标准主要起草人：姚健康、孔 宁、沈 烁。

# 引 言

当前，国家针对互联网应用大力开展了治理工作，作为处于互联网应用源头的域名注册服务，它在互联网的治理中处于极其重要的地位。只有在注册过程中实现域名的实名注册和数据安全，为域名应用的追溯提供可靠支撑，才能够为国家对互联网的管理提供有效的支持，实现后续网络服务的可管可控。

域名的注册过程包括若干环节，涉及注册服务机构、注册服务商、注册人等主体。因此，十分有必要针对域名注册服务中涉及的各方面技术问题制定相应的标准，从而使注册服务机构、注册代理、注册人等各相关主体能够遵循共同的规范，为域名注册系统的安全稳定运行提供可靠的参考和保证，进一步满足国家的互联网管理要求。

本标准主要规定了域名和主机的关系、主机的属性、EPP命令、正规语法以及和中文方面的规定。

# 域名注册协议主机供应技术要求

## 1 范围

本标准规定了对域名注册协议主机供应技术要求。

本标准适用于互联网相关的域名注册系统。

## 2 术语、定义和缩略语

### 2.1 术语和定义

下列术语和定义适用于本文件。

#### 2.1.1

**域名对象 Domain Object**

在EPP协议里的由英文字符或中文字符组成的域名代号，并具有一系列属性。

#### 2.1.2

**可扩展标记语言 Extensible Markup Language**

与HTML一样，都是标准通用语言。可扩展标记语言是互联网环境中跨平台的，依赖于内容的技术是当前处理结构化文档信息的有力工具。可扩展标记语言XML是一种简单的数据存储语言，使用一系列简单的标记规定数据。

#### 2.1.3

**主机对象 Host Object**

在EPP里用到表示具体机器的名字，并具有一系列属性。

#### 2.1.4

**联系人对象 Contact Object**

在EPP里用到的联系人代号，并具有一系列属性。

#### 2.1.5

**检测命令 Check**

EPP里用来查询对象是否存在的命令。

#### 2.1.6

**查询命令 Info**

EPP里用来查询对象的详细信息的命令。

#### 2.1.7

**创建命令 Create**

EPP里用来创建对象的命令。

#### 2.1.8

**删除命令 Delete**

EPP里用来删除对象的命令。

2.1.9

续费命令 Renew

EPP里用来给域名对象续费的命令。

2.1.10

转移命令 Transfer

EPP里用来在注册商之间转移域名对象的命令。

2.1.11

更新命令 Update

EPP里用来更新对象属性的命令。

2.2 缩略语

下列缩略语适用于本文件。

C	Client	客户端
DNS	Domain Name System	域名系统
EPP	Extensible Provisioning Protocol	可扩展供应协议
IANA	Internet Assigned Numbers Authority	互联网号牌分配机构
S	Server	服务器
UTC	Universal Coordinated Time	国际标准时间
XML	Extensible Markup Language	可扩展标志语言

3 协议概述

本标准规定了可扩展供应协议主机供应标准。这个标准采用可扩展标志语言 XML 及其框架格式作为基本格式。

4 协议要求

4.1 域名对象和主机对象的关系

本标准中如果域名对象对附属主机名对象有上级关系，比如域名“example.com”对主机名“ns1.example.com”有上级关系，EPP动作应保存这种上级关系。不保存这种关系的EPP动作（比如对象转换）必须被明确的拒绝。

在没有上级域名对象的库中，可以创建一个主机名对象。比如，可以在“.example”中创建主机名“ns1.example.com”，这样“.example”中的域名可以被授权给这个主机。在本标准中，如果主机被授权，但是主机名不属于这个命名空间，那么这样的主机被规定为“外部”的主机。一个主机是外部还是内部与这个主机用于授权的域有关。

4.2 对象属性

一个EPP主机对象有属性和相应的可以被客户端或者服务器端可见、可修改的属性值。这部分详实地规定了每个属性类型。

4.2.1 主机名

本标准要求域名和主机名应是中文域名或英文域名。



### 4.2.2 客户端标识

所有的EPP客户端被服务器带“cliDType”标志的唯一标识符标示。

### 4.2.3 状态值

一个主机对象必须总是至少有一个相应的状态值。状态值可能只被发起一个主机对象的客户端和对象所在的服务器一起设定。客户端可以用EPP的update命令更改主机对象的状态。

每个状态值可以伴随一串可读的规定对象状态的字符文本串。客户端禁止更改服务器设定的状态值。服务器可以按照本地服务器策略，修改或覆盖由客户端设定的状态值。一个对象的状态不是因为客户端发起的转换命令而更改，就是服务器操作员执行的行为而更改。

可以被客户端增删的状态值都在其前面加前缀“client”。相应地，服务器可以增删的状态值在其前面加“server”。在其前面不加“client”或“server”的状态值那一定是服务器管理的状态值。

状态值规定：

`clientDeleteProhibited, serverDeleteProhibited`

删除对象的任何请求都被拒绝。

`clientUpdateProhibited, serverUpdateProhibited`

更新对象（非删除）的请求必须拒绝。

`linked`

主机对象只有要与另外一个对象，比如一个域名对象，有一个活动的关联，服务器应该可以决定存在对象关联的服务。

`ok`

这是没有未完成的操作和禁令的对象的正常的状态值。就像增删别的状态值那样这个状态值也可被服务器增删。

`pendingCreate, pendingDelete, pendingTransfer, pendingUpdate`

转换命令提供给对象处理的（或者在<transfer>命令，对于主机对象的上级对象--域对象），但是行为没有被服务器完成。服务器操作员可以根据具体情况，例如允许人复查或者第三方行为，延迟动作完成。在处理但请求行为未完成的转换动作返回响应码1001。

当请求的行为已经完成，`pendingCreate`、`pendingDelete`、`pendingTransfer`，或者`pendingUpdate`状态值必须被删除。必须用“行为已经完成”或者“对象状态已经改变”的服务消息通知所有的事务中的客户端。

“ok”状态仅可以与“linked”状态合在一起用。

“linked”状态可以与任何状态联合。

“pendingDelete”状态一定不可以与`clientDeleteProhibited`和`serverDeleteProhibited`状态联合使用。

“pendingUpdate”状态一定不可以与`clientUpdateProhibited`和`serverUpdateProhibited`状态联合使用。

`pendingCreate`、`pendingDelete`、`pendingTransfer`和`pendingUpdate`的状态值一定不能互相联合使用。

其他的没有明确禁止的状态联合是允许使用的。

### 4.2.4 日期和时间

“日期和时间”属性值必须用国际标准时间。

#### 4.2.5 IP 地址

IPv4地址的语法和IPv6地址的语法应使用I正规语法。一个服务器可以拒绝使用还没有公开分配使用的IP地址。

### 4.3 EPP 命令映射

本标准的命令映射是专门用作通过EPP协议供应和管理互联网主机名。

#### 4.3.1 EPP 请求命令

EPP提供了两个获得主机信息的命令：`check`命令和`info`命令，`check`命令判断一台主机是否在一个库里，而`info`命令获得主机的详细信息。

##### 4.3.1.1 EPP<check>命令

EPP<check>命令用于决定是否能在一个库里供应一个对象。这个命令提供了允许客户获知供应对象成功或者失败的暗示。使用<create>命令作为对象供应要求最终是由服务器的策略决定的。

除了标准的EPP命令元素，<check>命令的子元素必须包含一个<host:check>元素，<host:check>元素用于标识域名空间。<host:check>元素包括如下子元素：

- 一个或者更多的<host:name>元素：包括需要查询的完全符合的主机对象。

具体例子见附录A.1。

当一个<check>命令被顺利执行后，EPP<resData>元素必须包括一个<host:chkData>子元素，这个子元素用来标识域名空间。在<host:chkData>元素中需要包括一个或者更多个<host:cd>元素。每个<host:cd>元素包括如下子元素：

- 一个<host:name>元素：包含所有符合条件的被检索的域对象。这个元素必须包含一个“avail”属性，该属性的值指示执行完<check>命令后对象的可用性（能否供应这个对象）。如果值为“1”或者“true”，那么表示能够供应对象；如果值为“0”或者“false”，那么表示不能供应这个对象。

- 一个可选的<dhost:reason>元素：如果不能供应对象时，可以提供这个元素。这个元素包含了服务器具体定义的文本，文本内容解释了对象不能供应的原因。在与客户协商之前，必须在相应语言中提供这个文本。当协议值不是缺省值“en”（英语）时，可以提供一个可选的“lang”属性来标志语言。

具体例子见附录A.2。

如果因为某种原因，不能执行<check>命令，那么必须返回EPP错误响应。

##### 4.3.1.2 EPP <info>命令

EPP<info>命令用于检索域对象的相关信息。除了标准的EPP命令元素，<info>命令必须包括一个<host:info>元素，用来标识命名空间。

具体例子请见附录A.3。

当一个<info>命令被成功执行后，EPP<resData>元素中必须包含一个<host:infData>子元素，这个子元素用来标识对象命名空间。<host:infData>元素包含如下子元素：

- 一个<host: name>元素：包含所有符合条件的主机对象名。
- 一个<host:roid>元素：包含创建对象时，赋给对象的库对象标识符。
- 0个或者更多可选的<host:status>元素：包含域名当前的状态规定。
- 0个或者更多的<host:addr>元素：包含主机相关联的IP地址。



- 一个<host:clID>元素：包含客户端的标识。
- 一个可选的<host:crID>元素：包含生成主机对象的标识。
- 一个<host:crDate>元素：包含主机对象生成时的日期和时间。
- 一个可选的<host:upID>元素：包含最后更新主机对象的客户端的标识符。如果主机对象从来没有修改过，这个元素就一定不能出现。
- 一个<host:upDate>元素：包含最新的主机对象更改的日期和时间。这个元素禁止用于从来没有执行过转换命令的主机对象。
- 一个<host:trDate>元素：包含最后一次成功执行主机对象transfer命令的日期和时间。这个元素禁止用于从来没有执行过转换命令的主机对象。

注意，主机一定不能直接传送。主机对象必须在主机的上级域传送时私下里传送。

当传送一个域对象时，易受传送影响的主机对象最后列在响应报文中，这个响应是在EPP域对象执行<info>命令之后返回的。

具体例子请见附录A.4。

如果<info>命令由于某种原因而不能执行，必须返回一个EPP错误响应。

#### 4.3.1.3 EPP <transfer>查询命令

EPP transfer命令不能直接应用在主机对象上，所以没有定义在EPP<transfer>查询命令映射。

#### 4.3.2 EPP 转换命令

EPP提供3个命令转换主机对象：<create>创建一个主机对象的实例，<delete>删除一个主机对象的实例，<update>改变一个主机对象相关的信息。本标准没有定义主机对象映射命令--EPP<renew>和<transfer>命令。

转换命令通常是实时进行和完成的。服务器操作符可能接受和处理转换命令，但是如果有人或者第三方提出查看请求，服务器可能延迟完成请求动作。在这种情况下，服务器必须向客户返回一个1001响应代码。这个代码就是告知客户命令已经被接收和执行，但是请求动作被挂起。服务器必须管理这个命令中主题对象的状态，用于反映发起和完成请求动作。一旦完成这个动作，必须向这个交易中涉及到的客户发送一个服务消息。消息的内容是动作已经完成，对象的状态改变了。除了要求的消息，也可以有其他告知方法。

服务器操作员应当确认一个客户是有权限对一个给定对象执行转换命令的。任何由未授权客户发起的转换对象的尝试都必须被拒绝。在这种情况下，服务器必须向客户返回一个2201响应，通知客户缺少权限执行请求命令。

##### 4.3.2.1 EPP <create> 命令

EPP<create>命令提供了一种允许客户端创建主机对象的转换操作。

除了标准的EPP命令元素，<create>命令必须包含一个<domain:create>元素，用于标识域名空间。<host:create>元素包含如下子元素：

- 一个<host: name>元素：包含完全符合条件的创建主机对象名。
- 0个或者更多<host:addr>元素：包含与主机的IP地址。每个元素可以包含：一个“ip”属性以说明IP地址格式。属性值“v4”被用作标识IPv4地址格式，而属性值“v6”被用作标识IPv6地址格式。如果“ip”属性没有指定，“v4”就是默认属性值。

主机可以被用作DNS系统的名字服务器。被用作域名服务器的主机可能易受服务器操作策略影响。这种策略需要或者禁止了IP地址。当用作域名服务器时，IP地址仅仅用来作为粘连记录。例如。如果一台“com”的权威服务器，服务器的名字是“ns1.example.net”，服务器就不需要生成服务器的DNS粘连记录了，服务器的ip地址也不需要了。如果一台主机名字存在于权威服务器的名字空间里，那么服务器必须在主机对象创建前，就知道这台主机的上级域。

具体例子请见附录A.5。

一个<create>命令成功执行，EPP<resData>元素必须包含一个标识主机名字空间的子元素。<host:creData>元素包含下列子元素：

- 一个<host:name>元素：包含完整合格的主机对象名。
- 一个<host:crData>元素：包含主机对象创建时的日期和时间。

具体例子请见附录A.6。

如果一个<create>命令由于某种原因不能执行，必须返回一个EPP错误响应。

4.3.2.2 EPP <delete> 命令

EPP<delete>命令提供了允许客户删除域对象的转换操作。除了标准的EPP命令元素，<delete>命令必须包含标识域名空间的<domain:delete>元素。<host:delete>元素包含如下子元素：

- 一个<domain: name>元素：包含完全符合条件的将要被删除的主机对象名。

如果主机对象与别的对象有关联，那么不能删除这个主机名对象。删除一个主机对象可能导致指向被删除的主机对象的域对象的DNS解析失败。比如，主机对象与域对象相关联，那么主机对象不应该被删除。

具体例子请见附录A.7。

当一个<delete>命令被成功执行之后，服务器必须用不带<resData>元素的EPP响应返回。具体例子请见附录A.8。

如果一个<delete>命令由于某种原因不能执行，必须返回一个EPP错误响应。

4.3.2.3 EPP <renew> 命令

EPP<renew>命令不能应用在主机对象上，所以没有为EPP<renew>命令定义的映射。

4.3.2.4 EPP <transfer> 命令

EPP<transfer>不能直接应用在主机对象上，所以没有为EPP<transfer>命令定义的映射。

4.3.2.5 EPP<update>命令

EPP<update>命令提供了一个允许客户修改域对象的属性的转换操作。除了标注的EPP命令元素，<update>命令必须包含一个标识命名空间的<host:update>元素。<host:update>元素包含如下子元素：

- 一个<domain: name>元素：包含完全符合条件的将要更新的主机对象名。
- 一个可选的<domain:add>元素：包含将要在对象中添加的属性值。
- 一个可选的<domain:rem>元素：包含要从对象删除的属性值。
- 一个可选的<domain:chg>元素：包含需要更改的属性值。

如果命令没有被扩展，那么至少要提供<host:add>、<host:rem>、<host:chg>元素中的一个。如果对<update>进行扩展，那么所有的元素可以忽略。

<host:add>和<host:rem>元素包含如下子元素：



— 0个或者更多的<domain:status>元素：包含要应用于对象或者从域对象删除的状态值。当具体制定某个值要删除时，只有属性值是重要的；

一个<host:chg>元素：包含下列子元素：

— 一个<host:name>元素：包含一个全新的完全合格的主机名。

主机名更改可能需要移除服务器已经接受的ip地址。IP地址联系可能容易受把主机作为服务器的服务器策略所影响。主机名更改可能对被称作主机对象的联合对象们产生一定的影响。一个主机名更改不应该需要额外的相关对象的更新以保存现有的联系，只有一个例外：更改一个外部主机对象。这个对象与不同的客户端发起的对象有联系。

为了直接更新这样的主机，就必须以最终返回EPP错误码2305而失败。这种改变可能用一个新名字和任何需要的新的属性而产生一个信的外部主机从而供应，结果更新了客户端发起的别的对象。

具体例子请见附录A.9。

当一个<update>命令被成功执行时，一个服务器响应一个没有<resData>元素的EPP消息。具体例子请见附录A.10。

如果<update>命令由于某种原因没有执行，那么必须返回一个EPP错误响应。

#### 4.3.3 请求动作的离线审查

服务器以从客户端收到命令的顺序来处理命令的。尽管一个立即的确认收到的响应和命令的处理是由服务器产生的，但是协议包括一些特性，这些特性允许在请求动作实际完成之前，对转换命令进行离线审查。在这种情况下，从服务器发来的响应必须清楚的告知命令已经被接收到和处理，但是请求动作被挂起。相应的对象状态必须清楚的显示这个挂起动作的处理。同样，当这个操作的离线处理完成时，服务器必须告知客户。具体例子请见附录A.11。

当返回给这个响应的域对象的状态必须包含“pendingCreate”。服务器操作员离线审查这个请求，然后通过使用<poll>命令检索服务消息告知客户审查的结果，或者使用带外机制告知客户请求。

服务消息必须包含规定<msg>元素中的通知内容，其中<msg>是响应<msgQ>元素中的子元素。此外，EPP<resData>元素必须包含一个<domain:panData>元素来标识命名空间。<domain:panData>包含如下子元素：

— 一个<host: name>元素：包含完全符合条件的域对象名。<host:name>元素包含一个指定的“paResult”属性。

— 一个正的布尔值代表请求被批准并且完成请求动作；一个负的布尔值代表请求被拒绝，没有执行请求动作。

— 一个<host:paTRID>元素：包含处理这个命令而返回的原始响应中的客户端事务标识符和服务器端事务标识符。客户端标识符是可选的，而且只有在客户端提供了原<creat>命令的标识符时才会返回标识符。

— 一个<domain:paDate>元素：包含规定审查请求动作的日期和时间。

具体例子请见附录A.12。

#### 4.4 形式语法

EPP对象映射在XML框架声明中具体说明。这里规定的语法是一个EPP的完整框架，该EPP是适用于自动验证EPP XML实例的。具体例子请见附录A.12。

4.5 中文化规定

EPP是用XML表示的，而XML支持用Unicode字符集及其的更紧凑的表示方法，例如UTF-8的编码信息。遵从XML的处理器既识别 UTF-8又识别UTF-16。虽然XML包含了一个条文，使之能够通过使用包含在一个 <?xml?>声明里的“encoding”属性，来标识和使用其他的编码字符编码，但是在解析器编码不兼容中文本地编码的情况下还是建议使用UTF-8。



# 附录 A

## (资料性附录)

### 例子

#### A.1

例: <check>命令:

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">

C: <command>

C: <check>

C: <host:check

C: xmlns:host="urn:ietf:params:xml:ns:host-1.0">

C: <host:name>ns1.example.com</host:name>

C: <host:name>ns2.example.com</host:name>

C: <host:name>ns3.example.com</host:name>

C: </host:check>

C: </check>

C: <clTRID>ABC-12345</clTRID>

C: </command>

C:</epp>

#### A.2

例: <check> 响应:

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">

S: <response>

S: <result code="1000">

S: <msg>Command completed successfully</msg>

S: </result>

S: <resData>

S: <host:chkData

S: xmlns:host="urn:ietf:params:xml:ns:host-1.0">

S: <host:cd>

S: <host:name avail="1">ns1.example.com</host:name>

S: </host:cd>

S: <host:cd>

S: <host:name avail="0">ns2.example2.com</host:name>

S: <host:reason>In use</host:reason>

S:       </host:cd>  
S:       <host:cd>  
S:        <host:name avail="1">ns3.example3.com</host:name>  
S:       </host:cd>  
S:       </host:chkData>  
S:       </resData>  
S:       <trID>  
S:        <clTRID>ABC-12345</clTRID>  
S:        <svTRID>54322-XYZ</svTRID>  
S:       </trID>  
S:       </response>  
S: </epp>

A.3

例: <info> 命令:  
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
C:   <command>  
C:    <info>  
C:     <host:info  
C:       xmlns:host="urn:ietf:params:xml:ns:host-1.0">  
C:        <host:name>ns1.example.com</host:name>  
C:       </host:info>  
C:     </info>  
C:     <clTRID>ABC-12345</clTRID>  
C:   </command>  
C:</epp>

A.4

例: <info> 响应:  
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
S:   <response>  
S:    <result code="1000">  
S:     <msg>Command completed successfully</msg>  
S:    </result>  
S:    <resData>  
S:     <host:infData

```

S:      xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S:      <host:name>ns1.example.com</host:name>
S:      <host:roid>NS1_EXAMPLE1-REP</host:roid>
S:      <host:status s="linked"/>
S:      <host:status s="clientUpdateProhibited"/>
S:      <host:addr ip="v4">192.0.2.2</host:addr>
S:      <host:addr ip="v4">192.0.2.29</host:addr>
S:      <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
S:      <host:clID>ClientY</host:clID>
S:      <host:crID>ClientX</host:crID>
S:      <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>
S:      <host:upID>ClientX</host:upID>
S:      <host:upDate>1999-12-03T09:00:00.0Z</host:upDate>
S:      <host:trDate>2000-04-08T09:00:00.0Z</host:trDate>
S:      </host:infData>
S:      </resData>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:      </response>
S: </epp>

```

## A.5

例: <create>命令:

Example <create> command:

```

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <host:create
C:        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:          <host:name>ns1.example.com</host:name>
C:          <host:addr ip="v4">192.0.2.2</host:addr>
C:          <host:addr ip="v4">192.0.2.29</host:addr>
C:          <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
C:        </host:create>

```

C:     </create>  
 C:     <clTRID>ABC-12345</clTRID>  
 C:     </command>  
 C: </epp>

## A.6

例: <create>响应:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 S:   <response>  
 S:     <result code="1000">  
 S:       <msg>Command completed successfully</msg>  
 S:     </result>  
 S:     <resData>  
 S:       <host:creData  
 S:         xmlns:host="urn:ietf:params:xml:ns:host-1.0">  
 S:           <host:name>ns1.example.com</host:name>  
 S:           <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>  
 S:         </host:creData>  
 S:     </resData>  
 S:     <trID>  
 S:       <clTRID>ABC-12345</clTRID>  
 S:       <svTRID>54322-XYZ</svTRID>  
 S:     </trID>  
 S:   </response>  
 S: </epp>

## A.7

例: <delete>命令:

C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 C:   <command>  
 C:     <delete>  
 C:       <host:delete  
 C:         xmlns:host="urn:ietf:params:xml:ns:host-1.0">  
 C:           <host:name>ns1.example.com</host:name>  
 C:         </host:delete>  
 C:     </delete>



C: <clTRID>ABC-12345</clTRID>  
 C: </command>  
 C:</epp>

## A.8

Example <delete> 响应:

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 S: <response>  
 S: <result code="1000">  
 S: <msg>Command completed successfully</msg>  
 S: </result>  
 S: <trID>  
 S: <clTRID>ABC-12345</clTRID>  
 S: <svTRID>54321-XYZ</svTRID>  
 S: </trID>  
 S: </response>  
 S:</epp>

## A.9

例: <update>命令:

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 C: <command>  
 C: <update>  
 C: <host:update  
 C: xmlns:host="urn:ietf:params:xml:ns:host-1.0">  
 C: <host:name>ns1.example.com</host:name>  
 C: <host:add>  
 C: <host:addr ip="v4">192.0.2.22</host:addr>  
 C: <host:status s="clientUpdateProhibited"/>  
 C: </host:add>  
 C: <host:rem>  
 C: <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>  
 C: </host:rem>  
 C: <host:chg>  
 C: <host:name>ns2.example.com</host:name>  
 C: </host:chg>

C:     </host:update>  
 C:     </update>  
 C:     <clTRID>ABC-12345</clTRID>  
 C:     </command>  
 C: </epp>

## A.10

例: <update> 响应:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 S:   <response>  
 S:     <result code="1000">  
 S:       <msg>Command completed successfully</msg>  
 S:     </result>  
 S:     <trID>  
 S:       <clTRID>ABC-12345</clTRID>  
 S:       <svTRID>54321-XYZ</svTRID>  
 S:     </trID>  
 S:   </response>  
 S: </epp>

## A.11

规定请求离线审查的<create>命令的例子如下。注意响应<create>命令的结果代码和消息:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 S:   <response>  
 S:     <result code="1001">  
 S:       <msg>Command completed successfully; action pending</msg>  
 S:     </result>  
 S:     <resData>  
 S:       <host:creData  
 S:         xmlns:host="urn:ietf:params:xml:ns:host-1.0">  
 S:         <host:name>ns1.example.com</host:name>  
 S:         <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>  
 S:       </host:creData>  
 S:     </resData>  
 S:     <trID>  
 S:       <clTRID>ABC-12345</clTRID>

S:     <svTRID>54322-XYZ</svTRID>  
 S:     </trID>  
 S:     </response>  
 S: </epp>

## A.12

例：“审查完成”的服务消息

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>  
 S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">  
 S:   <response>  
 S:     <result code="1301">  
 S:       <msg>Command completed successfully; ack to dequeue</msg>  
 S:     </result>  
 S:     <msgQ count="5" id="12345">  
 S:       <qDate>1999-04-04T22:01:00.0Z</qDate>  
 S:       <msg>Pending action completed successfully.</msg>  
 S:     </msgQ>  
 S:     <resData>  
 S:       <host:panData  
 S:         xmlns:host="urn:ietf:params:xml:ns:host-1.0">  
 S:         <host:name paResult="1">ns1.example.com</host:name>  
 S:         <host:paTRID>  
 S:           <clTRID>ABC-12345</clTRID>  
 S:           <svTRID>54322-XYZ</svTRID>  
 S:         </host:paTRID>  
 S:         <host:paDate>1999-04-04T22:00:00.0Z</host:paDate>  
 S:         </host:panData>  
 S:     </resData>  
 S:     <trID>  
 S:       <clTRID>BCD-23456</clTRID>  
 S:       <svTRID>65432-WXY</svTRID>  
 S:     </trID>  
 S:   </response>  
 S: </epp>

## A.13

BEGIN

```

<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:host-1.0"
  xmlns:host="urn:ietf:params:xml:ns:host-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <!--
  Import common element types.
  -->

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:epp-1.0"/>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      host provisioning schema.
    </documentation>
  </annotation>

  <!--
  Child elements found in EPP commands.
  -->

  <element name="check" type="host:mNameType"/>
  <element name="create" type="host:createType"/>
  <element name="delete" type="host:sNameType"/>
  <element name="info" type="host:sNameType"/>
  <element name="update" type="host:updateType"/>

  <!--
  Child elements of the <create> command.
  -->

  <complexType name="createType">

    <sequence>

```



```

    <element name="name" type="eppcom:labelType"/>
    <element name="addr" type="host:addrType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="addrType">
  <simpleContent>
    <extension base="host:addrStringType">
      <attribute name="ip" type="host:ipType"
        default="v4"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="addrStringType">
  <restriction base="token">
    <minLength value="3"/>
    <maxLength value="45"/>
  </restriction>
</simpleType>

<simpleType name="ipType">
  <restriction base="token">
    <enumeration value="v4"/>
    <enumeration value="v6"/>
  </restriction>
</simpleType>

<!--
Child elements of the <delete> and <info> commands.
-->

<complexType name="sNameType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
  </sequence>
</complexType>

```

&lt;!--

Child element of commands that accept multiple names.

--&gt;

&lt;complexType name="mNameType"&gt;

&lt;sequence&gt;

&lt;element name="name" type="eppcom:labelType"

maxOccurs="unbounded"/&gt;

&lt;/sequence&gt;

&lt;/complexType&gt;

&lt;!--

Child elements of the &lt;update&gt; command.

--&gt;

&lt;complexType name="updateType"&gt;

&lt;sequence&gt;

&lt;element name="name" type="eppcom:labelType"/&gt;

&lt;element name="add" type="host:addRemType"

minOccurs="0"/&gt;

&lt;element name="rem" type="host:addRemType"

minOccurs="0"/&gt;

&lt;element name="chg" type="host:chgType"

minOccurs="0"/&gt;

&lt;/sequence&gt;

&lt;/complexType&gt;

&lt;!--

Data elements that can be added or removed.

--&gt;

&lt;complexType name="addRemType"&gt;

&lt;sequence&gt;

&lt;element name="addr" type="host:addrType"

minOccurs="0" maxOccurs="unbounded"/&gt;

&lt;element name="status" type="host:statusType"

minOccurs="0" maxOccurs="7"/&gt;

&lt;/sequence&gt;

&lt;/complexType&gt;

<!--

Data elements that can be changed.

-->

```
<complexType name="chgType">
```

```
  <sequence>
```

```
    <element name="name" type="eppcom:labelType"/>
```

```
  </sequence>
```

```
</complexType>
```

<!--

Child response elements.

-->

```
<element name="chkData" type="host:chkDataType"/>
```

```
<element name="creData" type="host:creDataType"/>
```

```
<element name="infData" type="host:infDataType"/>
```

```
<element name="panData" type="host:panDataType"/>
```

<!--

<check> response elements.

-->

```
<complexType name="chkDataType">
```

```
  <sequence>
```

```
    <element name="cd" type="host:checkType"
```

```
      maxOccurs="unbounded"/>
```

```
  </sequence>
```

```
</complexType>
```

```
<complexType name="checkType">
```

```
  <sequence>
```

```
    <element name="name" type="host:checkNameType"/>
```

```
    <element name="reason" type="eppcom:reasonType"
```

```
      minOccurs="0"/>
```

```
  </sequence>
```

```
</complexType>
```

```
<complexType name="checkNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="avail" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!--
<create> response elements.
-->

<complexType name="creDataType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="crDate" type="dateTime"/>
  </sequence>
</complexType>

<!--
<info> response elements.
-->

<complexType name="infDataType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="roid" type="eppcom:roidType"/>
    <element name="status" type="host:statusType"
      maxOccurs="7"/>
    <element name="addr" type="host:addrType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="clID" type="eppcom:clIDType"/>
    <element name="crID" type="eppcom:clIDType"/>
    <element name="crDate" type="dateTime"/>
    <element name="upID" type="eppcom:clIDType"
      minOccurs="0"/>
    <element name="upDate" type="dateTime"/>
  </sequence>
</complexType>
```



```

    minOccurs="0"/>
    <element name="trDate" type="dateTime"
    minOccurs="0"/>
  </sequence>
</complexType>

<!--
Status is a combination of attributes and an optional human-readable
message that may be expressed in languages other than English.
-->
<complexType name="statusType">
  <simpleContent>
    <extension base="normalizedString">
      <attribute name="s" type="host:statusValueType"
        use="required"/>
      <attribute name="lang" type="language"
        default="en"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="statusValueType">
  <restriction base="token">
    <enumeration value="clientDeleteProhibited"/>
    <enumeration value="clientUpdateProhibited"/>
    <enumeration value="linked"/>
    <enumeration value="ok"/>
    <enumeration value="pendingCreate"/>
    <enumeration value="pendingDelete"/>
    <enumeration value="pendingTransfer"/>
    <enumeration value="pendingUpdate"/>
    <enumeration value="serverDeleteProhibited"/>
    <enumeration value="serverUpdateProhibited"/>
  </restriction>
</simpleType>

<!--

```

Pending action notification response elements.

-->

```
<complexType name="panDataType">
  <sequence>
    <element name="name" type="host:paNameType"/>
    <element name="paTRID" type="epp:trIDType"/>
    <element name="paDate" type="dateTime"/>
  </sequence>
</complexType>
```

```
<complexType name="paNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="paResult" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

<!--

End of schema.

-->

</schema>

END

## 参 考 文 献

- [1] IETF RFC1034 域名的概念与应用
  - [2] IETF RFC1035 域名的实现与规范
  - [3] IETF RFC1122 互联网主机传输层要求
  - [4] IETF RFC5732 主机供应技术要求
-

中华人民共和国  
通信行业标准  
域名注册协议主机供应技术要求  
YD/T 2421-2012

\*

人民邮电出版社出版发行  
北京市崇文区夕照寺街14号A座  
邮政编码：100061  
宝隆元（北京）印刷技术有限公司印刷  
版权所有 不得翻印

\*

开本：880×1230 1/16                      2013年3月第1版  
印张：2                                      2013年3月北京第1次印刷  
字数：50千字

15115·37

定价：25元

本书如有印装质量问题，请与本社联系 电话：(010)67114922