

XML DTD for Roaming Access Phone Book

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

This document defines the syntax as well as the semantics of the information to be included in the phone book for roaming applications. It comprises the information necessary to select the most appropriate ISP and to configure the host to get access to the network of the provider. The specification consists of a small set of required information elements and a variety of possible extensions. All data is specified in XML [5] (Extensible Markup Language) syntax leading to a concise XML DTD (Document Type Declaration) for the phone book.

Table of Contents

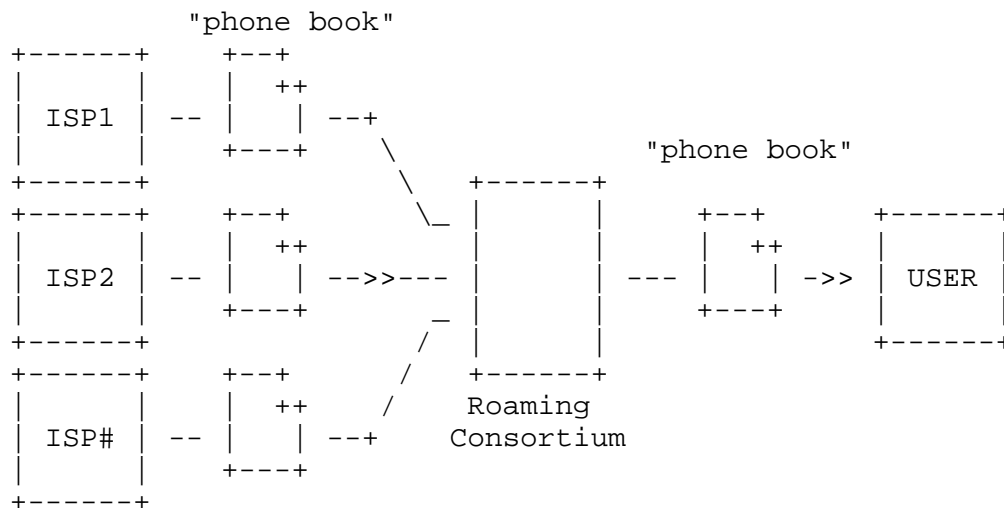
1. Introduction	3
2. Rationale for XML Usage	4
3. Specification of Requirements	5
4. Value type notations for 'stronger' typing	5
5. Container Element Definitions	5
5.1. PhoneBook	5
5.1.1. phoneBook Attribute "name"	6
5.1.2. phoneBook Attribute "version"	6
5.2. POP	7
5.2.1. pop Attribute "entryVersion"	8
5.3. Setup	8
5.4. Support	9
5.5. Provider	9

6.	Information Element Definitions	10
6.1.	Information elements defined for the POP element	10
6.1.1.	Address	10
6.1.1.1.	address Attribute "family"	10
6.1.1.2.	address Attribute "countryCode"	11
6.1.1.3.	address Attribute "areaCode"	11
6.1.2.	Media	11
6.1.2.1.	Modem Protocols	12
6.1.2.2.	ISDN Protocols	12
6.1.2.3.	ATM Protocols	13
6.1.2.4.	Frame Relay Protocols	13
6.1.2.5.	X.25 Protocols	13
6.1.3.	Minimum Data Rate	14
6.1.4.	Maximum Data Rate	14
6.1.5.	POP Properties	14
6.1.6.	Tunneling Protocols	15
6.1.7.	Dialing Script	15
6.1.8.	Pricing Information	16
6.1.9.	City	16
6.1.10.	Region	16
6.1.11.	Country	16
6.1.12.	POP Setup	17
6.1.13.	POP Support	17
6.1.14.	POP Provider	17
6.2.	Information elements defined for the Setup element ...	17
6.2.1.	DNS Server Address	17
6.2.2.	NNTP Server Name	18
6.2.3.	SMTP Server Name	18
6.2.4.	POP3 Server Name	18
6.2.5.	IMAP Server Name	18
6.2.6.	WWW Proxy	19
6.2.7.	FTP Proxy	19
6.2.8.	Winsock Proxy	19
6.2.9.	Default Gateway Address	19
6.2.10.	User Name Suffix	20
6.2.11.	User Name Prefix	20
6.3.	Information elements defined for the support element..	20
6.3.1.	Support Telephone Number	20
6.3.2.	Support Email Address	21
6.4.	Information elements defined for the provider element.	21
6.4.1.	Provider Name	21
6.4.2.	Provider Icon	21
6.4.3.	Provider's World Wide Web URL	21
6.4.4.	Provider's Main Email Address	22
6.4.5.	Billing Inquiry Email Address	22
6.4.6.	Further elements	22
7.	Complete XML DTD for the roaming phone book	22
8.	Security Considerations	28

9.	IANA Considerations	28
9.1.	Registration of new attribute values	29
9.2.	Registration of new information elements	29
10.	References	30
11.	Appendix: Examples	31
11.1.	The most simple example	31
11.2.	A more comprehensive example	31
12.	Acknowledgments	31
13.	Authors' Addresses	32
14.	Full Copyright Statement	33

1. Introduction

Roaming applications depend on the delivery of information about provided services and the procedures to get connected to the network from the roaming consortium to the individual users as well as from the operators of the network access servers, normally the members of the roaming consortium, and the roaming consortium.



The roaming consortium assembles from the individual contributions of the providers belonging to the consortium a unified version of the phone book for usage by the customers. Probably different groups of users get different versions of a phone book adapted to their particular needs. Even users might generate different subsets especially suited to particular applications from the information received from the roaming consortium, e.g., retrieving only entries for a particular country or extracting all access points providing wireless connectivity.

Therefore it is desirable to define a highly portable and well formed structure of the phone book to enable easy generation and postprocessing. Goals of this document include:

- Creating a flexible, extensible and robust framework upon which to build a standard phone book;
- Promoting a standard phone book format, to enhance interoperability between ISPs and roaming consortia as well as to enable automatic extraction of configuration data by a wide variety of devices;
- Defining a compact structure containing the essential information for the roaming user, to allow for storage and easy update even on small devices.

It is not intended by this document to create a plethoric solution, with phone book elements to fit every condition on earth, neither to define any kind of phone book update or transfer protocol.

2. Rationale for XML Usage

XML is rapidly becoming a standard format for data exchange between different applications also taking into account the transfer and access of data over the web. XML is used as syntax for expressing the structure and content of a roaming phone book to enable widespread usage and access to many different kind of media (e.g., paper, CDRom, www) using a widespread selection of access devices. Furthermore XML enables:

- Extensibility
- Flexibility
- Integration with directories

Extensibility is important because phone books are living documents; as such, it is unlikely that all the semantic requirements of arbitrary Internet service providers (ISPs) would be met by a fixed scheme, no matter how well thought out. Phone book designers must be free to create new attributes in a well-understood fashion to meet changing business needs.

Flexibility is required of the attribute definition syntax for many of the same reasons that semantic extensibility is necessary. If we assume that phone book designers may need to define elements of arbitrary type, the syntax chosen must be able to represent these data objects cleanly. Using XML for describing the data content of the phone book fits this bill nicely, since it can be used to unambiguously describe virtually any data type.

Integration with directories: although it is unlikely that phone books will be stored in the directory due to performance considerations, the creation of a XML DTD describing phone book content leaves that option open, with relatively little incremental effort required to implement it.

3. Specification of Requirements

In this document, the key words "MAY", "MUST", "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [1].

4. Value type notations for 'stronger' typing

XML DTDs do not currently have capabilities for 'strong typing' of the content of elements. The only type definition foreseen in the base specification is "#PCDATA", 'parsable character data'. This might be sufficient and is used throughout this document to define elements containing information mainly aimed for interpretation by human beings.

To enable a more concise description of the content of particular elements several value type notations are introduced. This allows for a more detailed type description of the content of elements in cases where it seems to be desirable.

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

<!-- Phone book value type notation declarations -->
<!NOTATION FQDN PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type Fully_qualified_domain_name">
<!NOTATION IPADR PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type IP_address">
<!NOTATION B64JPG PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type Base64_encoded_jpeg_image">
<!NOTATION B64GIF PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type Base64_encoded_gif_image">
```

5. Container Element Definitions

5.1. PhoneBook

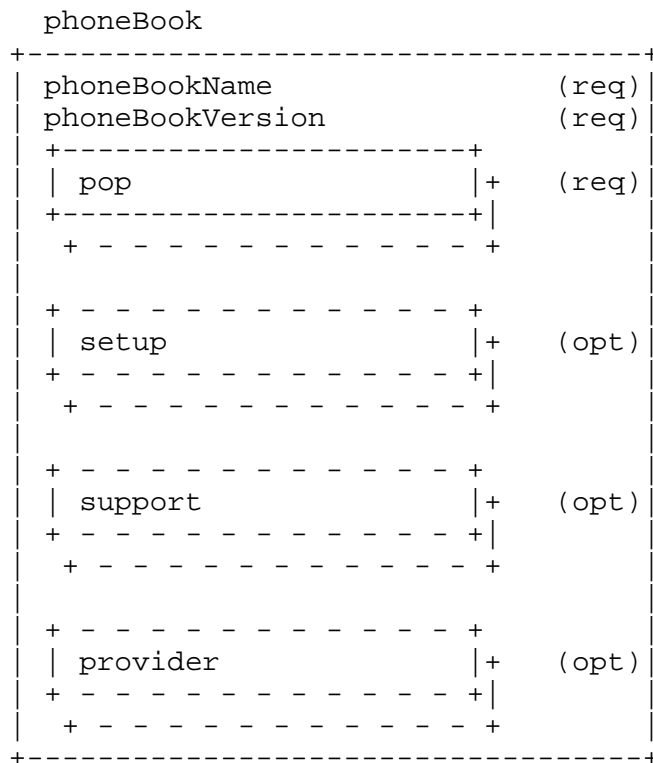
The phoneBook element is the basic container for phone book entries. It has two attributes, a phone book name and a phone book version number (applying to the phone book as a whole), and always contains one or more pop elements. A phoneBook element may also contain multiple Setup, Support and Provider elements, if they are referenced to by more than one pop element.

Syntax:

```

<!ELEMENT phoneBook (
    pop+,
    setup*,
    support*,
    provider*)>
<!--ATTLIST phoneBook
    name      CDATA      #REQUIRED
    version   CDATA      #REQUIRED -->

```



5.1.1. phoneBook Attribute "name"

The phoneBook attribute "name" is an arbitrary string assigned as an identifier for a phone book.

5.1.2. phoneBook Attribute "version"

The phoneBookVersion attribute is an integer representing the version of the phone book; it is a monotonically increasing counter which should be incremented each time the phone book is modified. This element can be used by a server to help decide what (if any) actions are required to bring a client's phone book up to date. For example, the client can, at connect time, send an update request to the server

including in the request the version number of its current phone book. If the client's phone book version is not the same as the server's current phone book version, the server can easily take appropriate action, e.g., reply with a URL pointing to a file containing the differences between the client and server phone books.

5.2. POP

The pop element contains information elements relevant to individual network points of presence (POPs). The required information elements are addrFamily, address, media and entryVersion. The media element represents the media types supported by the POP, while the entryVersion element is a monotonically-increasing integer which should be incremented whenever the object is modified.

The following information elements are currently defined for the pop element. Additional information elements may be defined by IANA in future.

POP	
entryVersion	(req)
address	(req)
media	(req)
minBitsPerSecond	(opt)
maxBitsPerSecond	(opt)
"popProperties"	(opt)
"tunnelingProtocols"	(opt)
dialScript	(opt)
pricingInformation	(opt)
"location"	(opt)
"popSetup"	(opt)
"popSupport"	(opt)
"popProvider"	(opt)

Syntax:

```
<!ENTITY % popInformation
    "address,
     media+,
     minBitsPerSecond?,
     maxBitsPerSecond?,
     popProperty*,
     tunnelProto*,
     dialScript?,
     pricingInformation?,
     city?,
     region?,
     country?,
     (setup | setupPtr)?,
     (support | supportPtr)?,
     (provider | providerPtr)?">

<!ELEMENT pop ( %popInformation; )>
<!ATTLIST pop
    entryVersion    CDATA    #REQUIRED>
```

5.2.1. pop Attribute "entryVersion"

The entryVersion attribute is an integer representing the version of the POP object; it is a monotonically increasing counter which should be incremented each time the object is modified. This attribute may be useful in merging and updating phone books.

5.3. Setup

The Setup element includes information elements which describe services which may change from provider to provider or even from POP to POP. Some of the values contained in these information elements may be available by other means (e.g., DHCP), but others may not.

The following information elements are currently defined for the Setup element. Additional information elements may be defined by IANA in future.

Syntax:

```
<!ENTITY % setupInformation
    "dnsServerAddress*,
     nntpServerName*,
     smtpServerName*,
     popServerName*,
     imapServerName*,
```



```

        wwwProxyServerName*,
        ftpProxyServerName*,
        winsockProxyServerName*,
        defaultGatewayAddress?,
        userNamePrefix?,
        userNameSuffix?">

<!ELEMENT setup ( %setupInformation; )>
<!--ATTLIST setup
        id          ID          #REQUIRED>

```

5.4. Support

The Support element includes those information elements that are pertinent to the provision of customer support for a POP or provider. Languages spoken by the staff at the support center might be specified by multiple entries for the attribute value language.

Additional information elements for the Support element may be defined by IANA in future.

Syntax:

```

<!ENTITY % supportInformation
        "(supportTelephoneNumber | supportMailtoURL)+">

<!--ELEMENT support %supportInformation; >
<!--ATTLIST support
        id          ID          #REQUIRED
        language    NMTOKENS    #IMPLIED >

```

5.5. Provider

The Provider element contains information elements pertaining to the general business operations of a given network service provider. The information elements include such things as telephone number, mailing address, etc., as well as URLs for e-mail and a World Wide Web site. A Provider element may also contain a reference to support information.

Currently the following information elements are defined for the Provider element. Additional information elements may be defined by IANA in future.

Syntax:

```

<!--ENTITY % providerInformation
        "providerName?,
        providerIcon?,
        wwwURL?,

```

```

generalMailtoURL?,
billingMailtoURL?,
businessCategory?,
x121Address?,
registeredAddress?,
destinationIndicator?,
preferredDeliveryMethod?,
telexNumber?,
teletexTerminalIdentifier?,
telephoneNumber?,
internationalISDNNumber?,
facsimileTelephoneNumber?,
street?,
postOfficeBox?,
postalCode?,
postalAddress?,
physicalDeliveryOfficeName?,
description?,
supportPtr*">

```

```

<!ELEMENT provider ( %providerInformation; )>
<!ATTLIST provider
    id          ID          #REQUIRED>

```

6. Information Element Definitions

6.1. Information elements defined for the POP element

6.1.1. Address

The address element provides the information representing the address of the POP. For POPs offering dial-up network access, the address element will at least contain an IA5 string representing a telephone number, formatted in standard fashion [4] (e.g., "+ 1 234 5678"). More detailed information may be available by optional attribute values.

Syntax:

```

<!-- A network address for this POP -->
<!ELEMENT address (#PCDATA)>

```

6.1.1.1. address Attribute "family"

The attribute family of the element address defines the address family to which the element value belongs. For POPs offering dial-up network access, the addrFamily attribute will generally contain a value for a telephone network based address family. Currently the following attribute values are defined. Additional values may be

registered by IANA in future.

Value	Description
-----	-----
E164	ITU-T E.164 (PSTN, SMDS, Frame Relay, ATM)
X121	ITU-T X.121 (X.25, Frame Relay)

Syntax:

```
<!-- Attribute values for address family -->
<!ENTITY % addressFamily "(E164|X121)" >
<!ATTLIST address
    family %addressFamily; #REQUIRED >
```

6.1.1.2. address Attribute "countryCode"

The countryCode attribute indicates the international dialing prefix for the country in which the POP is located.

Syntax:

```
<!-- ITU dialing code for the country in which this POP is located -->
<!ATTLIST address
    countryCode CDATA #IMPLIED >
```

6.1.1.3. address Attribute "areaCode"

The areaCode attribute contains the area or city code component of the telephone number in the 'address' element (if any) associated with this POP.

```
<!-- Area or city code component of the telephone number in the
    accessTelephoneNumber element associated with this POP -->
<!ATTLIST address
    areaCode CDATA #IMPLIED >
```

6.1.2. Media

The media element is a container describing the types of media and related protocols supported by this POP. The following media types are currently defined. Additional types may be registered by IANA in future.

Value	Media Type
-----	-----
viaMODEM	Modem
viaISDN	ISDN
viaATM	ATM
viaFR	Frame Relay
viaX25	X.25

Syntax:

```
<!-- The types of media supported by this POP -->
<!ENTITY % mediaTypes "(viaMODEM|viaISDN|viaATM|viaFR|viaX25)+" >
<!ELEMENT media %mediaTypes; >
```

6.1.2.1. Modem Protocols

The viaMODEM element is an empty element representing by its optional type attribute the modem protocol supported by the access devices that can be reached at address. To define multiple available protocols this element may be included repeatedly. The initially defined modem protocol types are listed in the table below. Additional values may be registered by IANA in future.

Value	Duplex	Speed	Protocol
-----	-----	-----	-----
V21	Full	300	ITU-T V.21
V22	Full	1200	ITU-T V.22
V29	Half	9600	ITU-T V.29
V32	Full	9600	ITU-T V.32
V32B	Full	14.4k	ITU-T V.32bis
V34	Full	28.8k	ITU-T V.34
V34B	Full	33.6k	ITU-T V.34bis
V90	Full	56k	ITU-T V.90

Syntax

```
<!-- A modem media type element -->
<!ENTITY % modemProtocols "(V21|V22|V29|V32|V32B|V34|V34B|V90)" >
<!ELEMENT viaMODEM EMPTY>
<!ATTLIST viaMODEM
    type %modemProtocols; #IMPLIED >
```

6.1.2.2. ISDN Protocols

The viaISDN element is an empty element representing by its optional type attribute the ISDN protocol supported by the access devices that can be reached at address. To define multiple available protocols this element may be included repeatedly. The initially defined ISDN protocol types are listed in the table below. Additional values may be registered by IANA in future.

Value	Speed	Meaning
-----	-----	-----
V110L	19.2k	ITU-T V.110
V110H	38.4k	ITU-T V.110
V120L	56k	ITU-T V.120

V120H	64k	ITU-T V.120
X75	64k	ITU-T X.75
HDLC	64k	RFC 1618

Syntax:

```
<!-- An ISDN media type element -->
<!ENTITY % isdnProtocols "(V110L|V110H|V120L|V120H|X75|HDLC)">
<!ELEMENT viaISDN EMPTY>
<!ATTLIST viaISDN
    type %isdnProtocols; #IMPLIED >
```

6.1.2.3. ATM Protocols

The viaATM element is an empty element representing by its optional type attribute a particular protocol supported by the access devices that can be reached at address. To define multiple available protocols this element may be included repeatedly. Currently only one protocol is defined. Additional values may be registered by IANA in future.

Syntax:

```
<!-- An ATM media type element -->
<!ENTITY % atmProtocols "(RFC2364)">
<!ELEMENT viaATM EMPTY>
<!ATTLIST viaATM
    type %atmProtocols; #IMPLIED >
```

6.1.2.4. Frame Relay Protocols

The viaFR element is an empty element representing by its optional type attribute the particular protocol supported by the access devices that can be reached at address. To define multiple available protocols this element may be included repeatedly. Currently only one protocol is defined. Additional values may be registered by IANA in future.

Syntax:

```
<!-- A Frame Relay media type element -->
<!ENTITY % frProtocols "(RFC1973)">
<!ELEMENT viaFR EMPTY>
<!ATTLIST viaFR
    type %frProtocols; #IMPLIED >
```

6.1.2.5. X.25 Protocols

The viaX25 element is an empty element representing by its optional type attribute the particular protocol supported by the access devices that can be reached at address. To define multiple available

protocols this element may be included repeatedly. Currently only one protocol is defined. Additional values may be registered by IANA in future.

Syntax:

```
<!-- A X.25 media type element -->
<!ENTITY % x25Protocols "(RFC1598)">
<!ELEMENT viaX25 EMPTY>
<!ATTLIST viaX25
    type %x25Protocols; #IMPLIED >
```

6.1.3. Minimum Data Rate

The minBitsPerSecond element indicates the minimum data rate (in bits/second) supported by the access devices at the POP.

Syntax:

```
<!-- Minimum data rate supported by this POP in bits/second -->
<!ELEMENT minBitsPerSecond (#PCDATA)>
```

6.1.4. Maximum Data Rate

The maxBitsPerSecond element indicates the maximum data rate (in bits/second) supported by the access devices at the POP.

Syntax:

```
<!-- Maximum data rate supported by this POP in bits/second -->
<!ELEMENT maxBitsPerSecond (#PCDATA)>
```

6.1.5. POP Properties

The popProperty element is an empty element representing by its attribute value a particular property of this POP. To define multiple available protocols this element might be included several times. The initially defined properties are listed in the table below. Additional values may be registered by IANA in future.

Value	Property
-----	-----
MPPP	Multilink PPP (RFC 1990)
MOBIP	Mobile IP (RFC 2002)
MCRX	Multicast Reception
MCTX	Multicast Transmission

Syntax:

```
<!-- A property characterizing this POP -->
<!ENTITY % popProperties "(MPPP|MOBIP|MCRX|MCTX)" >
```

```

<!ELEMENT popProperty EMPTY>
<!ATTLIST popProperty
    type %popProperties; #REQUIRED>

```

6.1.6. Tunneling Protocols

The tunnelProto element is an empty element representing by its attribute a tunneling protocol supported by this POP. To define multiple available protocols this element might be included several times. The initially defined values are listed in the table below. Additional values may be registered by IANA in future.

Value	Protocol	
-----	-----	
L2TP	RFC 2661	L2TP
PPTP	RFC 2637	PPTP
L2F	RFC 2341	L2F
ATMP	RFC 2107	ATMP
AHT	RFC 2402	IP AH Tunnel Mode
ESPT	RFC 2406	IP ESP Tunnel Mode
IPIP	RFC 1853	IP-IP
MIP	RFC 2004	Minimal IP-IP
GRE	RFC 1701	GRE

Syntax:

```

<!-- A tunneling protocol supported by this POP -->
<!ENTITY % tunnelingProtocols
    "(L2TP|PPTP|L2F|ATMP|AHT|ESPT|IPIP|MIP|GRE)" >
<!ELEMENT tunnelProto EMPTY>
<!ATTLIST tunnelProto
    type %tunnelingProtocols; #REQUIRED>

```

6.1.7. Dialing Script

The dialScript element contains the dialing script to be used when connecting to this POP. The attribute value type of dialScript defines the type of the script that should be used when connecting to this POP.

Syntax:

```

<!-- The dial script to be used -->
<!ELEMENT dialScript (#PCDATA)>
<!ATTLIST dialScript
    type CDATA #IMPLIED >

```

6.1.8. Pricing Information

The pricingInformation element is a free-form string representing pricing information for this POP. It may be anything from a simple string indicating relative expense (e.g., "\$\$\$\$" for a very expensive POP) to a paragraph describing time-of-day and other differential pricing variables.

Syntax:

```
<!-- Pricing information for this POP -->
<!ELEMENT pricing (#PCDATA)>
```

6.1.9. City

The city element contains the name of the city in which the POP is located (not the city(s) from which it is accessible by a local call).

Syntax:

```
<!-- The name of the city in which this POP is located -->
<!ELEMENT city (#PCDATA)>
```

6.1.10. Region

The region element contains the name of the region in which the POP is located. In the United States, this would be the name of a state or (for Washington, D.C.) administrative district. In other countries, it might be the name of a province, parish or county.

Syntax:

```
<!-- The name of the region in which this POP is located -->
<!ELEMENT region (#PCDATA)>
```

6.1.11. Country

The country element contains the name of the country in which the POP is located. The country name may be abbreviated (e.g., "USA" for the United States of America or "UK" for the United Kingdom) but if abbreviations are used the usage must be consistent within a given phone book.

Syntax:

```
<!-- The name of the country in which this POP is located -->
<!ELEMENT country (#PCDATA)>
```


6.1.12. POP Setup

The popSetup element is either a setup element, if setup is specific to this particular POP, or a reference to any of the setup elements given in the outer scope of the phonebook element.

Syntax:

```
<!-- Reference for setup information for this POP -->
<!ELEMENT setupPtr EMPTY>
<!ATTLIST setupPtr
    setupID    IDREFS    #IMPLIED>
```

6.1.13. POP Support

The popSupport element is either a support element, if support is specific to this particular POP, or a reference to any of the support elements given in the outer scope of the phonebook element.

Syntax:

```
<!-- Reference for support information for this POP -->
<!ELEMENT supportPtr EMPTY>
<!ATTLIST supportPtr
    supportID  IDREFS    #IMPLIED>
```

6.1.14. POP Provider

The popProvider element is either a provider element, if provider information is specific to this particular POP, or a reference to any of the provider elements given in the outer scope of the phonebook element.

Syntax:

```
<!-- Reference for provider information for this POP -->
<!ELEMENT providerPtr EMPTY>
<!ATTLIST providerPtr
    providerID IDREFS    #IMPLIED>
```

6.2. Information elements defined for the Setup element

6.2.1. DNS Server Address

The dnsServerAddress element represents the IP address of the Domain Name Service (DNS) server which should be used when connected to this POP. The address is represented in the form of a string in dotted-decimal notation (e.g., 192.168.101.1).

Syntax:

```
<!-- Domain Name Server IP address -->
<!ELEMENT dnsServerAddress (#PCDATA)>
<!ATTLIST dnsServerAddress
    value NOTATION (IPADR) #IMPLIED>
```

6.2.2. NNTP Server Name

The nntpServerName element contains the fully qualified domain name (FQDN) of the Network News Transfer Protocol (NNTP) server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an NNTP server -->
<!ELEMENT nntpServerName (#PCDATA)>
<!ATTLIST nntpServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.3. SMTP Server Name

The smtpServerName element contains the FQDN of the Simple Mail Transfer Protocol (SMTP) server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an SMTP mail server -->
<!ELEMENT smtpServerName (#PCDATA)>
<!ATTLIST smtpServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.4. POP3 Server Name

The popServerName element contains the FQDN of the Post Office Protocol (POP) server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an POP3 mail server -->
<!ELEMENT popServerName (#PCDATA)>
<!ATTLIST popServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.5. IMAP Server Name

The imapServerName element contains the FQDN of the Internet Mail Access Protocol (IMAP) server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an IMAP4 server -->
<!ELEMENT imapServerName (#PCDATA)>
<!ATTLIST imapServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.6. WWW Proxy

The `wwwProxyServerName` element contains the FQDN of the World Wide Web (WWW) proxy server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an WWW Proxy -->
<!ELEMENT wwwProxyServerName (#PCDATA)>
<!ATTLIST wwwProxyServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.7. FTP Proxy

The `ftpProxyServerName` element contains the FQDN of the File Transfer Protocol (FTP) proxy server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an FTP Proxy -->
<!ELEMENT ftpProxyServerName (#PCDATA)>
<!ATTLIST ftpProxyServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.8. Winsock Proxy

The `winsockProxyServerName` element contains the FQDN of the Windows Socket (Winsock) proxy server which should be used when connected to this POP.

Syntax:

```
<!-- Name of an Winsock Proxy -->
<!ELEMENT winsockProxyServerName (#PCDATA)>
<!ATTLIST winsockProxyServerName
    value NOTATION (FQDN) #IMPLIED>
```

6.2.9. Default Gateway Address

The `defaultGatewayAddress` element represents the address of the default gateway which should be used when connected to this POP. The address is represented in the form of a string in dotted-decimal notation (e.g., 192.168.101.1).

Syntax:

```
<!-- Default Gateway IP address (in dotted decimal notation) -->
<!ELEMENT defaultGatewayAddress (#PCDATA)>
<!ATTLIST defaultGatewayAddress
    value NOTATION (IPADR) #IMPLIED>
```

6.2.10. User Name Suffix

The `userNameSuffix` element represents a string which should be concatenated to the base username. For example, if the base username is "userA" and the value of this element is "@bigco.com", the resulting augmented username would be "userA@bigco.com". An intelligent dialer may concatenate the string automatically. Note that both the `userNameSuffix` and the `userNamePrefix` (below) may be applied to the same base username.

Syntax:

```
<!-- User Name suffix -->
<!ELEMENT userNameSuffix (#PCDATA)>
```

6.2.11. User Name Prefix

The `userNamePrefix` element represents a string to which the base username should be concatenated. For example, if the base username is "userB" and the value of this element is "BIGCO/" the resulting augmented username would be "BIGCO/userB". An intelligent dialer may perform the concatenation automatically. Note that both the `userNameSuffix` (above) and the `userNamePrefix` may be applied to the same base username.

Syntax:

```
<!-- User Name prefix -->
<!ELEMENT userNamePrefix (#PCDATA)>
```

6.3. Information elements defined for the support element

6.3.1. Support Telephone Number

The `supportTelephoneNumber` element contains a number that may be called to reach the support center for a particular provider or POP. This element is basically a string and should contain the entire telephone number in international form, e.g., "+1 425 838 8080".

Syntax:

```
<!-- The number to be dialed to contact customer support
    for this POP or provider -->
<!ELEMENT supportTelephoneNumber (#PCDATA)>
```

6.3.2. Support Email Address

The supportMailtoURL element contains a URL for the provider's customer support email address, e.g., mailto:support@uu.net. This URL could be used to contact customer support personnel regarding non-urgent issues.

Syntax:

```
<!-- A Uniform Resource Locator for the provider's customer
      support email address -->
<!ELEMENT supportMailtoURL (#PCDATA)>
```

6.4. Information elements defined for the provider element

6.4.1. Provider Name

The providerName element is a string containing the name of the provider (e.g., "BIGNET Corporation").

Syntax:

```
<!-- The name of the provider -->
<!ELEMENT providerName (#PCDATA)>
```

6.4.2. Provider Icon

The providerIcon attribute contains a BASE64 encoded JPEG or GIF image which may be used for 'branding' phone book entries or displayed when dialing.

Syntax:

```
<!-- An icon in BASE64 encoded JPEG or GIF format -->
<!ELEMENT providerIcon (#PCDATA)>
<!ATTLIST providerIcon
      value NOTATION (B64JPG | B64GIF) #IMPLIED>
```

6.4.3. Provider's World Wide Web URL

The wwwURL element contains a Uniform Resource Locator (URL) for the provider's Web site, for example, http://www.uu.net.

Syntax:

```
<!-- A Uniform Resource Locator for the provider's home page -->
<!ELEMENT wwwURL (#PCDATA)>
```

6.4.4. Provider's Main Email Address

The `generalMailtoURL` element contains a URL for the provider's main email address, for example, `mailto:contact@uu.net`. This URL could be used for general correspondence, complaints, etc.

Syntax:

```
<!-- A Uniform Resource Locator for the provider's
      email address -->
<!ELEMENT generalMailtoURL (#PCDATA)>
```

6.4.5. Billing Inquiry Email Address

The `billingMailtoURL` element contains a URL for the provider's billing support email address, for example, `mailto:billing@uu.net`. This URL could be used to for correspondence regarding billing and payment issues.

Syntax:

```
<!-- A Uniform Resource Locator for the email
      address to be used for billing inquiries -->
<!ELEMENT billingMailtoURL (#PCDATA)>
```

6.4.6. Further elements

The remainder of the information elements of the provider element are described in principle in [3].

7. Complete XML DTD for the roaming phone book

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

<!-- Parameter entity declaration -->
<!-- ++++++ -->
<!-- This section will be maintained by IANA and can be direct
      referenced by the DTD specification by means of an external
      parameter entity. -->

<!ENTITY % addressFamily "(E164|X121)" >

<!ENTITY % mediaTypes "(viaMODEM|viaISDN|viaATM|viaFR|viaX25)+" >

<!ENTITY % modemProtocols "(V21|V22|V29|V32|V32B|V34|V34B|V90)" >

<!ENTITY % isdnProtocols "(V110L|V110H|V120L|V120H|X75|HDLC)">

<!ENTITY % atmProtocols "(RFC2364)">
```

```
<!ENTITY % frProtocols "(RFC1973)">

<!ENTITY % x25Protocols "(RFC1598)">

<!ENTITY % popProperties "(MPPP|MOBIP|MCRX|MCTX)" >

<!ENTITY % tunnelingProtocols
    "(L2TP|PPTP|L2F|ATMP|AHT|ESPT|IPIP|MIP|GRE)" >

<!ENTITY % popInformation
    "address,
    media+,
    minBitsPerSecond?,
    maxBitsPerSecond?,
    popProperty*,
    tunnelProto*,
    dialScript?,
    pricingInformation?,
    city?,
    region?,
    country?,
    (setup|setupPtr)?,
    (support|supportPtr)?,
    (provider|providerPtr)?">

<!ENTITY % setupInformation
    "dnsServerAddress*,
    nntpServerName*,
    smtpServerName*,
    popServerName*,
    imapServerName*,
    wwwProxyServerName*,
    ftpProxyServerName*,
    winsockProxyServerName*,
    defaultGatewayAddress?,
    userNamePrefix?,
    userNameSuffix?">

<!ENTITY % supportInformation
    "(supportTelephoneNumber|supportMailtoURL)+">

<!ENTITY % providerInformation
    "providerName?,
    providerIcon?,
    wwwURL?,
    generalMailtoURL?,
    billingMailtoURL?,
    businessCategory?,
```

```

    x121Address?,
    registeredAddress?,
    destinationIndicator?,
    preferredDeliveryMethod?,
    telexNumber?,
    teletexTerminalIdentifier?,
    telephoneNumber?,
    internationalISDNNumber?,
    facsimileTelephoneNumber?,
    street?,
    postOfficeBox?,
    postalCode?,
    postalAddress?,
    physicalDeliveryOfficeName?,
    description?,
    supportPtr*">

<!-- ++++++ End of IANA maintained section ++++++ -->

<!-- Phone book value type notation declarations -->
<!NOTATION FQDN PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type Fully_qualified_domain_name">
<!NOTATION IPADR PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type IP_address">
<!NOTATION B64JPG PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type Base64_encoded_jpeg_image">
<!NOTATION B64GIF PUBLIC "-//IETF/roamPhoneBook/NOTATION
value Type Base64_encoded_gif_image">

<!-- Phone book element declarations -->
<!ELEMENT phoneBook (
    pop+,
    setup*,
    support*,
    provider*) >
<!ATTLIST phoneBook
    name      CDATA      #REQUIRED
    version   CDATA      #REQUIRED >

<!ELEMENT pop ( %popInformation; )>
<!ATTLIST pop
    entryVersion   CDATA      #REQUIRED>

<!ELEMENT setup ( %setupInformation; )>
<!ATTLIST setup
    id             ID          #REQUIRED>

<!ELEMENT support ( %supportInformation; )>

```



```
<!-- ATTLIST support
      id          ID          #REQUIRED
      language    NMTOKENS    #IMPLIED >

<!-- ELEMENT provider ( %providerInformation; )>
<!-- ATTLIST provider
      id          ID          #REQUIRED>

<!-- Information elements for pop -->
<!-- ELEMENT address (#PCDATA)>
<!-- ATTLIST address
      family      %addressFamily; #REQUIRED
      countryCode CDATA            #IMPLIED
      areaCode    CDATA            #IMPLIED >

<!-- ELEMENT media %mediaTypes; >

<!-- ELEMENT viaMODEM EMPTY>
<!-- ATTLIST viaMODEM
      type %modemProtocols; #IMPLIED >

<!-- ELEMENT viaISDN EMPTY>
<!-- ATTLIST viaISDN
      type %isdnProtocols; #IMPLIED >

<!-- ELEMENT viaATM EMPTY>
<!-- ATTLIST viaATM
      type %atmProtocols; #IMPLIED >

<!-- ELEMENT viaFR EMPTY>
<!-- ATTLIST viaFR
      type %frProtocols; #IMPLIED >

<!-- ELEMENT viaX25 EMPTY>
<!-- ATTLIST viaX25
      type %x25Protocols; #IMPLIED >

<!-- ELEMENT minBitsPerSecond (#PCDATA)>

<!-- ELEMENT maxBitsPerSecond (#PCDATA)>

<!-- ELEMENT popProperty EMPTY>
<!-- ATTLIST popProperty
      type      %popProperties; #REQUIRED >

<!-- ELEMENT tunnelProto EMPTY>
<!-- ATTLIST tunnelProto
      type      %tunnelingProtocols; #REQUIRED >
```

```
<!-- ELEMENT dialScript (#PCDATA) -->
<!-- ATTLIST dialScript
      type CDATA #IMPLIED -->

<!-- ELEMENT pricing (#PCDATA) -->

<!-- ELEMENT city (#PCDATA) -->

<!-- ELEMENT region (#PCDATA) -->

<!-- ELEMENT country (#PCDATA) -->

<!-- ELEMENT setupPtr EMPTY -->
<!-- ATTLIST setupPtr
      setupID IDREFS #IMPLIED -->

<!-- ELEMENT supportPtr EMPTY -->
<!-- ATTLIST supportPtr
      supportID IDREFS #IMPLIED -->

<!-- ELEMENT providerPtr EMPTY -->
<!-- ATTLIST providerPtr
      providerID IDREFS #IMPLIED -->

<!-- Information elements for setup -->
<!-- ELEMENT dnsServerAddress (#PCDATA) -->
<!-- ATTLIST dnsServerAddress
      value NOTATION (IPADR) #IMPLIED -->

<!-- ELEMENT nntpServerName (#PCDATA) -->
<!-- ATTLIST nntpServerName
      value NOTATION (FQDN) #IMPLIED -->

<!-- ELEMENT smtpServerName (#PCDATA) -->
<!-- ATTLIST smtpServerName
      value NOTATION (FQDN) #IMPLIED -->

<!-- ELEMENT popServerName (#PCDATA) -->
<!-- ATTLIST popServerName
      value NOTATION (FQDN) #IMPLIED -->

<!-- ELEMENT imapServerName (#PCDATA) -->
<!-- ATTLIST imapServerName
      value NOTATION (FQDN) #IMPLIED -->

<!-- ELEMENT wwwProxyServerName (#PCDATA) -->
<!-- ATTLIST wwwProxyServerName
      value NOTATION (FQDN) #IMPLIED -->
```

```
<!ELEMENT ftpProxyServerName (#PCDATA)>
<!ATTLIST ftpProxyServerName
    value NOTATION (FQDN) #IMPLIED>

<!ELEMENT winsockProxyServerName (#PCDATA)>
<!ATTLIST winsockProxyServerName
    value NOTATION (FQDN) #IMPLIED>

<!ELEMENT defaultGatewayAddress (#PCDATA)>
<!ATTLIST defaultGatewayAddress
    value NOTATION (IPADR) #IMPLIED>

<!ELEMENT userNameSuffix (#PCDATA)>

<!ELEMENT userNamePrefix (#PCDATA)>

<!-- Information elements for support -->
<!ELEMENT supportTelephoneNumber (#PCDATA)>

<!ELEMENT supportMailtoURL (#PCDATA)>

<!-- Information elements for provider -->
<!ELEMENT providerName (#PCDATA)>

<!ELEMENT providerIcon (#PCDATA)>
<!ATTLIST providerIcon
    value NOTATION (B64JPG|B64GIF) #IMPLIED>

<!ELEMENT wwwURL (#PCDATA)>

<!ELEMENT generalMailtoURL (#PCDATA)>

<!ELEMENT billingMailtoURL (#PCDATA)>

<!-- Further provider elements according to RFC1274 -->

<!ELEMENT businessCategory (#PCDATA)>

<!ELEMENT x121Address (#PCDATA)>

<!ELEMENT registeredAddress (#PCDATA)>

<!ELEMENT destinationIndicator (#PCDATA)>

<!ELEMENT preferredDeliveryMethod (#PCDATA)>

<!ELEMENT telexNumber (#PCDATA)>
```

```
<!ELEMENT teletexTerminalIdentifier (#PCDATA)>
<!ELEMENT telephoneNumber (#PCDATA)>
<!ELEMENT internationalISDNNumber (#PCDATA)>
<!ELEMENT facsimileTelephoneNumber (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT postOfficeBox (#PCDATA)>
<!ELEMENT postalCode (#PCDATA)>
<!ELEMENT postalAddress (#PCDATA)>
<!ELEMENT physicalDeliveryOfficeName (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!-- end of dtd -->
```

8. Security Considerations

The secure distribution and transport of information of a phone book for roaming applications require a reliable authentication of the issuer of the information as well as means to preserve the integrity of the provided information.

No specific elements for security requirements are provided by the phone book XML DTD itself. It is assumed that security of the roaming phone book is provided by means outside of the scope of this specification, such as signing the phone book using pgp.

9. IANA Considerations

This specification provides the possibility to define further attribute values for all information elements owning enumerated attribute lists as well as to extend the main structures 'pop', 'setup', 'support' and 'provider' by additional information elements. Therefore the specification of the roaming phone book can be adopted to future requirements without changing this document. Extensions and refinements to this specification can be achieved by registration of new elements and attributes by IANA.

Extending this specification with additional attributes or elements must not change the validity of documents based on an older version of the XML DTD. Therefore all added information elements must be

optional, prohibiting the mandatory inclusion of newly defined information elements. Adding new values to enumerated attribute lists has no backward compatibility constraints because it does not harm the validity of attributes already defined.

To facilitate the registration of new information elements and attribute values the DTD of the phone book has been separated in two parts, the extensible part containing only parameter entity declarations for ease inclusion of new values, and the fixed part containing the detailed specification of the content and structure of the phone book. By referencing the parameter entity declarations in the fixed part of the specification the whole phone book becomes extensible.

The part containing the parameter entity declarations has to be maintained by the IANA. There are two different classes of declarations in this part requiring different policies for registering new values.

9.1. Registration of new attribute values

The entities 'addressFamily', 'modemProtocols', 'isdnProtocols', 'atmProtocols', 'frProtocols', 'x25Protocols', 'popProperties' and 'tunnelingProtocols' are describing enumerated attribute value lists. Because there is no limitation in the name space of these attribute values and newly defined attribute values can not harm the validity of existing values, new attribute values can be assigned by Specification Required [6].

9.2. Registration of new information elements

The entities 'mediaTypes', 'popInformation', 'setupInformation', 'supportInformation' and 'providerInformation' define the information elements probably included in the media, pop, setup, support and provider elements. Inserting new values into these lists extends the phone book by arbitrarily new information elements. Inappropriate use of the XML content model can destroy the backward compatibility of the DTD. Therefore the assignment of new information elements requires the approval of a Designated Expert [6]. In addition to the insertion of a new value into the list, the detailed definition of the information element has to be appended to the specification part maintained by the IANA.

10. References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Reynolds, J. and J. Postel, "ASSIGNED NUMBERS", STD 2, RFC 1700, October 1994.
- [3] Barker, P. and S. Kille, "The COSINE and Internet X.500 Schema", RFC 1274, November 1991.
- [4] ITU Rec. E.123, "Notation for national and international telephone numbers", 1988.
- [5] "Extensible Markup Language (XML) 1.0" W3C Recommendation 10-February-1998 <http://www.w3.org/TR/1998/REC-xml-19980210>
- [6] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

11. Appendix: Examples

11.1. The most simple example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE phoneBook SYSTEM "roamPhoneBook.dtd">
<phoneBook name="minimalExample" version="1">
  <pop entryVersion="1">
    <address family="E164">+1 234 5678901</address>
    <media><viaMODEM/></media>
  </pop>
</phoneBook>
```

11.2. A more comprehensive example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE phoneBook SYSTEM "roamPhoneBook.dtd">
<phoneBook name="KNF_simple" version="1">
  <pop entryVersion="1">
    <address family="E164" countryCode="49">+49913130540</address>
    <media>
      <viaMODEM type="V90"/>
      <viaMODEM type="V34B"/>
      <viaISDN type="HDLN"/>
    </media>
    <setup>
      <dnsServerAddress>192.168.147.5</dnsServerAddress>
      <dnsServerAddress>193.175.24.33</dnsServerAddress>
    </setup>
  </pop>
  <support id="KNF_main" language="EN DE">
    <supportMailtoURL>mailto:support@franken.de</supportMailtoURL>
    <supportTelephoneNumber>+499123968066</supportTelephoneNumber>
  </support>
</phoneBook>
```

12. Acknowledgments

Thanks to Pat Calhoun, Bernard Aboba, Jay Farhat, Butch Anton, Quentin Miller, and Ken Crocker for salient input and review.

13. Authors' Addresses

Questions about this memo can be directed to:

Max Riegel
Siemens AG
Hofmannstr. 51
Munich, 81359
Germany

Phone: +49 89 722 49557
EMail: maximilian.riegel@icn.siemens.de

Glen Zorn
Cisco Systems, Inc.
500 108th Avenue N.E., Suite 500
Bellevue, WA 98004
USA

Phone: +1 425 438 8218
EMail: gwz@cisco.com

14. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

