



**IPDR
Service Specification
Design Guide**

Version 3.5.1

November, 2004

© 1999-2004 IPDR.org, Inc.

Preface

Contacts

For general questions regarding this document and referrals to technical experts for detailed questions, please contact:

Chief Editor: Steve Cotton
Cotton Management Consulting
editor@ipdr.org

Acknowledgements

The following member companies contributed materially to the creation of this document:

Charter Members



SBC Laboratories, Inc.



Supporting Members

[American Management Systems](#)
[CIBERNET](#)
[CSG Systems](#)
[Dan Net A/S](#)
[EDB](#)
[Telescience AS](#)
[GRIC Communications, Inc.](#)
[Infosys Technologies](#)
[Intel Corporation](#)
[Intrado](#)
[Marconi](#)
[NEC America](#)
[Openet](#)
[Swisscom IT](#)

[Services Syniverse Technologies, Inc.](#)
[TeleStrategies, Inc.](#)
[VeriSign](#)

Associate Members

[Broadband Communications Solutions Corp.](#)
[Canadian Clearinghouse](#)
[CPqD](#)
[Digital Route](#)
[MetraTech Corp.](#)
[Nomadix, Inc.](#)
[TTogewanet](#)

Abstract

The IP-based services discussed in the abstract in the IPDR Business Applications Requirements must be documented explicitly for each individual service supported by OSS/BSS products that implement IPDR protocols. The document that embodies the specification of a given service is known as an IPDR Service Specification. This document serves as a guide for those authoring and using such Service Specifications.

Change History

1.0 Initial Draft	November 19, 2003
1.0 Ballot Draft	October 22, 2001
1.0	November 19, 2003
3.5 Initial Draft	July 10, 2003
3.5 Review Draft 1	December 12, 2004
3.5	April 25, 2004—Production Release
3.5.1	November 4, 2004—Add macAddress to derived types, provide guidelines for ordering of schema elements.

Table of Contents

Preface.....	2
Contacts	2
Acknowledgements	2
Abstract.....	3
Change History	3
1. Introduction.....	5
1.1 Purpose.....	5
1.2 Scope.....	5
1.3 Compatibility	5
1.4 Overview.....	5
2. Service Specification Template	6
3. Use Case Guidelines	7
4. Data Definition Guidelines	8
5. Service Definition Guidelines.....	9
5.1 Schema Partitioning	9
5.2 Element Specification	9
5.3 Element Typing.....	10
5.4 Enumerations	12
5.5 Structures	13
5.6 Annotations of element definitions.....	14
5.7 Composition of schemas to create service definitions.....	15
5.8 Schema references in IPDRDoc instance documents	16

1 Introduction

1.1 Purpose

This document, in conjunction with the referenced Service Definition documents, is intended to specify technical information that is sufficient for practical implementations of interchange of usage data among service elements participating in the delivery of IP-based services, either within a single enterprise or across multiple enterprises.

The IPDR organization (IPDR.org) intends to submit this specification to selected accredited organizations for consideration as an approved standard.

1.2 Scope

This document is limited to the discussion of issues as defined by the mission statement of IPDR.org, namely:

The IPDR Organization (the “Organization”) is organized and operates as a non-stock not for profit organization for the following purposes:

- To create and promote the adoption of interoperability standards for exchanging service usage and control information between IP network or hosting elements and operations or business support systems.
- To provide a standardized framework for the development of carrier-grade support systems that enable next-generation digital service providers to operate efficiently and cost effectively.

1.3 Compatibility

Future revisions are expected to make every attempt to preserve investments made by service providers and solution vendors by considering backward and forward compatibility whenever it is practical.

1.4 Overview

This balance of this document is divided into four major chapters:

- Use Case Guidelines—requirements and direction for the business model of the service specified.
- Data Definition Guidelines—requirements for the form and style of usage attribute data descriptions.
- XML Schema Guidelines—constraints on the use of the XML Schema notation and specific requirements for layout and style.

2 Service Specification Template

Service Specification documents shall conform to the following outline:

- Title Page – in the style of the title page of this document
- Preface – containing subsections contained in this document
- Introduction – containing subsections contained in this document, modified as appropriate to the specific service
- Use Case – a business process description of one or more service usage scenarios, as viewed from the perspective of the Service Consumer and Service Provider
- Data Definitions – a list of the usage attributes determined to be essential to the application described in the Use Case
- Service Definition – the annotated XML schema document implementing the packaging of the usage attributes in an IPDR-compliant document

3 Use Case Guidelines

The preferred representation of use cases for IPDR Service Specification is UML. All components of UML that apply to the business scenario(s) described are encouraged to be used. The use cases should be expressed from the perspective of the Service Consumer, namely the events and metrics described should be in terms of those visible to the Service Consumer and of relevance to them with respect to service quality and units of usage. In cases where there are industry standards and practices describing a service, deference should be made to these sources. The IPDR Service Specification is not intended to diverge from industry standards and practices, but rather focus on the unique aspect of defining the IPDR recorded by an IR from usage data reflecting the actions of the Service Consumer.

4 Data Definition Guidelines

The data attributes documented in the Service Specification should be derived from analysis of the use case(s) presented in the prior section. Events and metrics generated by the Service Element, in response to the service consumed by the Service Consumer should be enumerated in a tabular representation as follows:

Category	Name	Type	Presence (Required, Optional, Conditional)	Permitted Values	Remarks
<p><i>One of the choices from the high-level architecture model, namely: Who When What Where Why</i></p>	<p><i>A unique name describing the attribute. Every attempt should be made to reuse names from prior Service Specifications, thus promoting the use of common data formats and minimizing the casual proliferation of names.</i></p>	<p><i>One of the types specified in the following section. Note that the analyst need not enumerate such subsidiary elements as timeZoneOffset or currency codes, as these will be assumed by the designer of the service definition and supplied in the XML Schema information model.</i></p>	<p><i>One of the choice: Required-the IPDR is not complete without this attribute present Optional-the IPDR can be considered complete without this attribute present Conditional-this attribute is required to be present if another attribute is present (or attains a specified value) <u>or</u> this attribute must attain a specified value (or values) if another attribute takes on a specified value (or values).</i></p>	<p><i>An enumeration of specific values (if applicable) that the attribute may or must take on, possible under conditions specified in the prior column. Reference to applicable and normative third-party standards should be explicitly noted here.</i></p>	<p><i>Supporting explanation of particular considerations that were borne in mind as a result of the analysis of the use case(s).</i></p>

5. Service Definition Guidelines

This section describes guidelines that should be followed when writing IPDR service definitions in order to support the effective mapping to both the UTF8 string-based encoding and compact encoding forms.

5.1 Schema Partitioning

A minimal IPDR service definition involves at least three separate schema definition files:

- IPDRTypes.xsd - the set of types and annotation elements used in all IPDR based definitions, specified in XML Schema format.
- IPDRDocX.Y.xsd - defines the XML structure of an IPDR document in XML Schema format. [Note: X.Y denotes the major and minor revision levels of the IPDR schema document defining the IPDRDoc structure.]
- <service><version>.xsd - an XML schema description of the elements associated with a particular service and their type. In addition it specifies accounting record types indicating whether elements must appear in an IPDR record. A service should use the XML schema <import> mechanism on the previous two schema definitions.

As new services are defined by IPDR, new service definitions are defined in XML Schema format. Service definitions may be created within IPDR.org working groups or may be defined by third parties. All service definitions should use a unique name space to clearly separate them.

5.2 Element Specification

The service definitions must separate the element specification from the complex type definition that specifies the content of accounting records. Although this is not required by XML-Schema, it will aid in the processing of service definition files that are also utilized by the compact encoding format.

It should be noted that some XML parsers rely on all element definitions being specified before any use of that element in a reference. This dictates that such a practice be used to ensure maximal compatibility. It should also be noted that the elements referenced in complexType declarations should be in the same order as the primary specification of the elements.

An example explicit element definition would appear as:

```
<element name="movieId" type="string">
  <annotation>
    <documentation>
      An identifier used to uniquely identify this movie.
    </documentation>
  </annotation>
</element>
```

Element definitions must not be present in complex type definitions. When describing the contents of an IPDR record, ref attributes must be used.

Correct:

```
<complexType name="Vod-IPDR">
  <extension base="ipdr:IPDR">
    <sequence>
      <element ref="movieId" minOccurs="1"/>
      <element ref="movieName" minOccurs="1"/>
      ...
    </sequence>
  </extension>
</complexType>
```

Incorrect:

```
<complexType name="Vod-IPDR">
  <extension base="ipdr:IPDR">
    <sequence>
      <!-- name is used rather than ref -->
      <element name="movieId" type="string"/>
      <element ref="movieName" minOccurs="1"/>
      ...
    </sequence>
  </extension>
</complexType>
```

Element definitions must not contain any attributes. Previous IPDR service specifications had some elements that used attributes to specify units of measure. The units are now considered fixed for a given element definition, and should be made explicit by using the "ipdr:units" application annotation mechanism described later.

5.3 Element Typing

Element specifications must all derive from the XML Simple types. Complex types are not allowed. This restriction is necessary to preserve the mapping to the compact format. Today all service definitions utilize simple types.

The types allowed are further constrained to be the subset of base data types defined below, enumeration based restrictions of strings (which use the ipdr:enumid annotation) or the simple types defined as part of IPDRTypes.xsd.

The following XML-Schema simple data types are allowed in IPDR Service definitions:

- string - for non-numeric data which does not have a more specific IPDR specialized type. The XML-Schema type string should be used. The compact format will encode this as an octet array containing the UTF-8 encoding of this string.
- integer or int - either type is acceptable, however, int is preferred. In both cases the compact encoding will assume that this is a signed 32-bit integer quantity.

- long - this type should be used for integer quantities which may exceed 2^{31} and are less than 2^{63} (~9 quintillion). Values that may exceed this value or have a fractional component should be represented as a float or double.
- float - the compact encoding will represent this quantity as a 32-bit floating variable.
- double - the compact encoding will represent this quantity as a 64-bit floating variable.
- dateTime - the compact encoding will represent this quantity as a 32-bit integer representing the seconds since EPOCH (Jan 1, 1970 0:00 GMT). It is further constrained to always use the timezone designator "Z" indicating GMT. Note that all dateTime elements (including the IPDR type extensions of dateTime below) should be accompanied, if applicable) by an integer element named timeZoneOffset which indicates the number of minutes ahead/behind GMT that represents the local time.

The additional two base types are allowed but discouraged:

- unsignedInt - this type is allowed but discouraged. It is represented as value between 0 and 2^{32} . If the additional positive integers beyond that provided by the int type are to be used it is recommended to use the long type.
- unsignedLong - this type is allowed but discouraged. It only differs from long in its ability to represent values between 2^{63} and 2^{64} (9-18 quintillion). If these values are desired, the use of double is recommended.

In addition to these basic types, IPDR defines the following type extensions. Any additional types must be defined by IPDR, and will be added to the IPDRTypes.xsd schema definition.

- ipdr:dateTimeMsec - this type supports time resolution at the millisecond level. Its compact representation uses a 64-bit integer quantity to represent the msec since EPOCH (Jan 1, 1970 0:00 GMT). It is further constrained to always use the timezone designator "Z" indicating GMT. Quantities of this type can optionally use 3 digits of fraction after the second to represent the milliseconds. If absent it is assumed the millisecond component is ".000".
- ipdr:dateTimeUsec - this type is supports time resolution at the microsecond level. Its compact representation uses a 64-bit integer quantity to represent the Usec since EPOCH (Jan 1, 1970 0:00 GMT). It is further constrained to always use the timezone designator "Z" indicating GMT. Quantities of this type can optionally use 6 digits of fraction after the second to represent the microseconds. If absent it is assumed the millisecond component is ".000000".
- ipdr:ipV4Addr - an IP version 4 address in dotted notation decimal (e.g., 15.13.120.22). This is represented in the compact encoding as an unsigned 32-bit integer. This form is

used rather than octet strings because it is expected to be common and requires 4 vs. 8 bytes to encode.

- ipdr:ipV6Addr - an IPv6 address in colon separated 2 byte block hexadecimal notation (e.g., FEDC:AB19:12FE:0234:98EF:1178:8891:CAFF). This is represented in the compact encoding as a 16 byte octet string.
- ipdr:ipAddr – a non-specific version of an IP address (either v4 or v6), of type hexBinary, determined as to version by the application through lexical parsing. This is represented in the compact encoding as a 16 byte octet string.
- ipdr:uuid - a universal unique id in hex dash notation (e.g.f81d4fae-7dec-11d0-a765-00a0c91e6bf6). This is represented in the compact encoding as a 16 byte octet string.
- ipdr:macAddress - An IEEE 802®-2001 Medium Access Control address in hyphen separated 1 byte block hexadecimal notation. Example: 0F-2C-D0-44-2E-09.

Currency elements should be encoded as a triple of amount, exponent (placement of the decimal point in the amount), and currency type (denoted in ISO 4217:2001 string form).

Additional type definitions must be registered through IPDR.org, and will be reflected in updated versions of IPDRTypes.xsd. For all other types, string should be used, and the compact format will also encode these as a string.

5.4 Enumerations

Enumerations are allowed in service definitions. They must derive from the base class of "integer". In addition each enumeration item specified may contain an "ipdr:enumMeaning" annotation element which provides the textual equivalent.

If such an annotation is present, the compact representation will be represented as a 32-bit integer containing the appropriate numeric value.

Example:

```
<element name="completionCode">
  <simpleType>
    <restriction base="integer">
      <enumeration value="1">
        <annotation>
          <documentation>
            Indicates this operation completed successfully.
          </documentation>
          <appinfo>
            <ipdr:enumMeaning>Success </ipdr:enumMeaning>
          </appinfo>
        </annotation>
      </enumeration>
    </restriction>
  </simpleType>
</element>
```

```
</enumeration>
<enumeration value="2">
  <annotation>
    <documentation>
      Indicates the line was in use.
    </documentation>
    <appinfo>
      <ipdr:enumMeaning>Busy</ipdr:enumMeaning>
    </appinfo>
  </annotation>
</enumeration>
<enumeration value="3">
  <annotation>
    <documentation>
      Indicates the call was terminated without answer.
    </documentation>
    <appinfo>
      <ipdr:enumMeaning>No Answer</ipdr:enumMeaning>
    </appinfo>
  </annotation>
</enumeration>
</restriction>
</simpleType>
</element>
```

5.5 Structures

Below is an example schema fragment demonstrating substructures and repeating elements.

```
<element name="volume" type="int"/>
<element name="qosLevel" type="int"/>
<element name="startTime" type="dateTime"/>
<element name="endTime" type="dateTime"/>

<complexType name="qosVolumeEvent">
  <sequence>
    <element ref="volume"/>
    <element ref="qosLevel"/>
    <element ref="startTime"/>
    <element ref="endTime"/>
  </sequence>
</complexType>

<element name="qosVolume" type="qosVolumeEvent"/>

<element name="subscriber" type="string"/>
```

```
<element name="sourceStationID" type="string"/>

<complexType name="GatewayExample">
  <complexContent>
    <extension base="ipdr:IPDR">
      <element ref="subscriber"/>
      <element ref="sourceStationID"/>
      <element ref="qosVolume" maxOccurs="unbounded"/>
    </extension>
  </complexContent>
</complexType>
```

An example IPDR record in XML format following this structure would appear as:

```
<IPDR xsi:type="GatewayExample">
  <subscriber>k123455</subscriber>
  <sourceStationID>station7.region3.foo.com</sourceStationID>
  <qosVolume>
    <volume>100</volume>
    <qosLevel>2</qosLevel>
    <startTime>20020904T13:13:13Z</startTime>
    <endTime>20020904T13:15:13Z</endTime>
  </qosVolume>
  <qosVolume>
    <volume>200</volume>
    <qosLevel>5</qosLevel>
    <startTime>20020904T13:15:14Z</startTime>
    <endTime>20020904T13:19:13Z</endTime>
  </qosVolume>
</IPDR>
```

5.6 Annotations of element definitions

The XML-Schema document describing a service definition should constitute a standalone document. As such the use of annotations is critical in conveying the intent of each attribute defined for a service.

XML-Schema provides for documentation based annotation as well as application specific annotation that can consist of additional elements.

All elements defined as part of a service definition must provide a documentation element that describes the meaning of this element.

In addition to the textual documentation, the following application based annotation may appear:

- `ipdr:units` - defines the unit of measure associated with this element.
- `ipdr:reference` - specifies a URL which provides additional (human readable) information about this element.
- `ipdr:status` - takes on the value "current" or "deprecated".

Note that as service definitions are updated, elements should never be simply removed. If it is determined that they should not be used in future revisions, they should be marked as deprecated.

Example:

```
<element name="greenStampCount" type="integer">
  <annotation>
    <documentation>
      The number of Green Stamps earned for using this service.
      Note that green stamps are no longer issued, as of
      Jan. 30, 2001.
    </documentation>
    <appinfo>
      <ipdr:units>one stamp</ipdr:units>
      <ipdr:reference>http://www.greenstamps.com/stamps</ipdr:reference>
      <ipdr:status>deprecated</ipdr:status>
    </appinfo>
  </annotation>
</element>
```

5.7 Composition of schemas to create service definitions

An IPDR service definition that adds to the IPDR namespace will be constructed using the `<include>` and `<import>` mechanisms of XML-Schema.

An IPDR service definition should use an alternate namespace to protect from naming collisions. In this case the service definition will specify a different target namespace in its initial declaration. To incorporate the appropriate IPDR base named types, the `<import>` mechanism of XML-Schema should be used as follows:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ipdr = "http://www.ipdr.org/public/IPDR"
  targetNamespace = "http://www.foo.com/ipdr/namespace"
  xmlns:foo = http://www.foo.com/ipdr/namespace
  version = "3.5-A.0"
  elementFormDefault = "qualified"
  attributeFormDefault = "unqualified">
<include schemaLocation = "http://www.ipdr.org/public/IPDRTypes.xsd"/>
```

```
<import namespace="http://www.ipdr.org/public/IPDR"  
  schemaLocation="http://www.ipdr.org/public/IPDRDoc3.5.xsd"/>  
</schema>
```

5.8 Schema references in IPDRDoc instance documents

When defining an instance document, the namespace and schemaLocation should be specified as follows:

```
<IPDRDoc xmlns="http://www.ipdr.org/public/namespaces "  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.ipdr.org/public/foo.xsd">
```

If there were multiple services in a single document, then the header would appear as:

```
<IPDRDoc xmlns="http://www.ipdr.org/public/namespaces"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.ipdr.org/public/namespaces  
    http://www.ipdr.org/public/service1.xsd  
    http://www.ipdr.org/public/namespaces  
    http://www.ipdr.org/public/service2.xsd">
```

If non-IPDR defined service definitions were used in the document, then the header would appear as:

```
<IPDRDoc xmlns="http://www.ipdr.org/public/namespaces"  
  xmlns:foo="http://www.foo.com/ipdr/namespaces"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.foo.com/ipdr/namespaces  
    http://www.foo.com/ipdr/fooService.xsd">
```