

International Telecommunication Union

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**J.204**

(06/2008)

SERIES J: CABLE NETWORKS AND TRANSMISSION  
OF TELEVISION, SOUND PROGRAMME AND OTHER  
MULTIMEDIA SIGNALS

Application for Interactive Digital Television

---

## **Metrics gathering specification**

Recommendation ITU-T J.204





# **Recommendation ITU-T J.204**

## **Metrics gathering specification**

### **Summary**

Recommendation ITU-T J.204 defines a data format and transmission protocol for the collection of service delivery metrics on customer premises equipment, such as digital cable set top boxes

### **Source**

Recommendation ITU-T J.204 was approved on 13 June 2008 by ITU-T Study Group 9 (2005-2008) under Recommendation ITU-T A.8 procedure.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2009

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## CONTENTS

	<b>Page</b>
1 Scope .....	1
1.1 Introduction and Purpose.....	1
2 References.....	1
2.1 Normative References .....	1
2.2 Informative References .....	1
3 Terms and Definitions .....	2
4 Abbreviations and acronyms .....	2
5 Overview .....	2
5.1 General Context.....	2
6 Metrics Requirements.....	3
6.1 Transmission Protocol.....	3
6.2 Data Model.....	5
Annex A – Metrics Data Model Schema .....	7
Annex B – Schema Data Element Requirements.....	16
B.1 protocolVersionMajor .....	16
B.2 protocolVersionMinor .....	16
B.3 hostID .....	16
B.4 hostType .....	16
B.5 createTime .....	16
B.6 transmitTime.....	16
B.7 privateData .....	16
B.8 serviceSelection.....	16
B.9 tuning.....	17
B.10 SDV .....	17
B.11 servicePresentation .....	17
B.12 segmentPresentation .....	17
B.13 DPI.....	17
B.14 inputEvent.....	18
B.15 applicationLoadRequested .....	18
B.16 applicationLoaded .....	18
B.17 applicationLifecycle .....	18
B.18 environmentSelection.....	18
B.19 applicationMode .....	19
B.20 application .....	19



# Recommendation ITU-T J.204

## Metrics gathering specification

### 1 Scope

This Recommendation defines a data format and transmission protocol for the collection of service delivery metrics on customer premises equipment, such as digital cable set top boxes. Metrics are records generated by relevant service delivery events, such as a service selection event. Each metrics record includes a timestamp of the event. This technology provides a means to accurately and efficiently collect metrics for the purpose of measuring service delivery.

NOTE – The structure and content of this Recommendation have been organized for ease of use by those familiar with the original source material; as such, the usual style of ITU-T Recommendations has not been applied.

#### 1.1 Introduction and Purpose

The purpose of this Recommendation is to specify metrics collection and transmission mechanisms and interfaces on receiver platforms. The data formats and transmission protocols may also be adopted by proprietary systems.

For the purposes of this Recommendation, the term "metrics" refers to specific data, defined herein, that is generated by a receiver and transmitted to a cable headend.

Metrics, as defined in this Recommendation, are intended to complement other measurement systems that may be employed by cable operators.

### 2 References

#### 2.1 Normative References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ATIS] Usage Data Management For Packet-Based Services – Service-Neutral Protocol Specification For Billing Applications, ATIS-0300075.1.2006, August 2006.  
<https://www.atis.org/docstore/product.aspx?id=22606>
- [IPDR/SP] IPDR/SP Protocol Specification Version 2.2, August 25, 2006.  
<http://www.ipdr.org/public/DocumentMap/SP2.2.pdf>
- [IPDR/XDR] IPDR/XDR File Encoding Format, Version 3.5.1, November 2004.  
<http://www.ipdr.org/public/DocumentMap/XDR3.5.1.pdf>

#### 2.2 Informative References

- [OCAP 1.1] OpenCable Applications Platform, Profile 1.1, OC-SP-OCAP1.1-I01-061229, December 29, 2006, Cable Television Laboratories, Inc.  
<http://www.opencable.com/downloads/specs/OC-SP-OCAP1.1-I01-061229.pdf>
- [ETV] Enhanced TV Binary Interchange Format, OC-SP-ETV-BIF1.0-I04-070921, September 21, 2007, Cable Television Laboratories, Inc.  
<http://www.cablelabs.com/specifications/OC-SP-ETV-BIF1.0-I04-070921.pdf>

### 3 Terms and Definitions

This Recommendation defines the following terms:

**3.1 authorized collector:** An Event Tracking API-compliant server that implements the receiving side of the IPDR Streaming Protocol, and which has been authorized to participate in the overall Collection System.

**3.2 data message:** A message transmitted between an IPDR Exporter and Collector across the Streaming Protocol, containing a common Streaming Protocol header and an optional payload consisting of control and Data Records.

**3.3 data record:** The binary encoding of an IPDR record.

**3.4 Internet Protocol Detail Record (IPDR):** The fundamental unit of data transferred between an Exporter and a Collector. It is defined by a Template and contains Fields.

**3.5 IPDR-Type:** A constraint on the value and format of an individual Field within a Data Record; e.g., dateTime.

**3.6 Internet Protocol Detail Record/Streaming Protocol (IPDR/SP):** The protocol used to transfer Data Messages and Data Records between Exporter and Collectors.

**3.7 IPDR/SP collector functionality:** An implementation on the data-receiving side of the IPDR/SP. It enables the reception and collection of Data Records from IPDR/SP Exporters. It is typically part of an OSS/BSS, or a mediation system.

**3.8 IPDR/SP exporter functionality:** An implementation on the data-producing side of the IPDR/SP. It enables formatting and sending of Data Records to an interested consumer system, e.g., at a cable headend using the IPDR/SP.

### 4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

CPE Customer Premises Equipment

IPDR Internet Protocol Detail Record

IPDR/SP Internet Protocol Detail Record Streaming Protocol

### 5 Overview

#### 5.1 General Context

Metrics are a set of well-defined data that are reported to a cable network by a receiver, for the purpose of measuring certain aspects of cable service delivery. This Recommendation specifies a data model and transmission protocol. Platforms that support this Recommendation are expected to define the semantics for the data points that are syntactically defined herein.

The means for metrics collection on the receiver is composed of a log, which contains log records. A log and log records are created and stored in a format determined by an implementer. Log records are created by the receiver in response to well-defined events, such as operations performed by a receiver on behalf of applications, or generated directly by applications. A log and its log records are transmitted from the receiver in a well-known format, under control of logic implemented by the network operator.

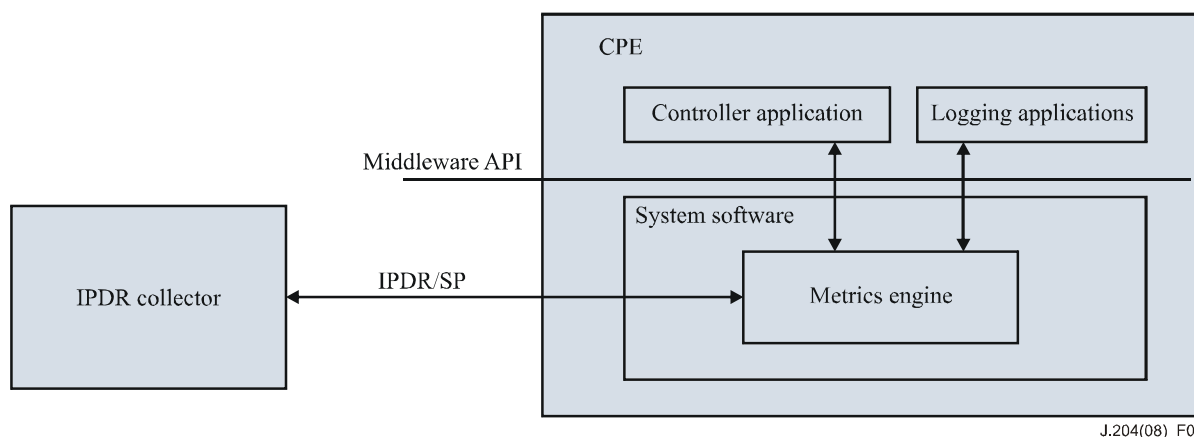


There are three logical metrics components: a so-called "metrics engine" implemented within a receiver; one or more privileged applications that configure and manage the metrics engine, collectively called the "metrics controller"; and any other "metrics logging" applications that interface with the metrics engine to generate application-specific log records.

This Recommendation defines requirements for a metrics engine and the interface between a metrics engine and a cable headend. Metrics data is considered private to the cable network. Functions such as storage, aggregation, and reporting are the responsibility of a network operator. This interface provides a well-known data format and protocol to ensure that metrics are comparable across cable networks.

The metrics engine is a component of the system software that includes the operating system and middleware, such as OCAP that communicates with the controller and logging applications as well as the cable headend (see Transmission Protocol clause below). The middleware (e.g., OCAP), is a logical software layer between the operating system and the applications that abstracts the functionality of the hardware and other functions and provides an interoperable application programming interface layer.

The controller application informs the metrics engine which metrics data are of interest



**Figure 1 – Logical Model of Metrics System**

Interfaces between metrics controller and logging applications and the metric engine are currently not defined within this Recommendation. We expect to include definition of a Java API for this purpose in a subsequent revision to this Recommendation.

The metrics engine incorporates IPDR/SP Exporter functionality. A cable headend that supports metrics implements IPDR/SP Collector functionality. See <https://www.atis.org/docstore/product.aspx?id=22606>

[IPDR/SP] for details.

## 6 Metrics Requirements

This clause defines requirements for metrics and the interface between a metrics engine and a cable headend.

### 6.1 Transmission Protocol

IPDR/SP is the protocol for transmitting metrics data to a cable headend. It is an American National Standard ATIS-0300075.1.2006 [ATIS]. The IPDR/SP utilizes a light-weight, session-oriented, real-time streaming protocol to quickly and reliably deliver metrics data, a.k.a. IPDRs, from an "Exporter" to a "Collector" (typically a type of mediation system). Receivers that support this

metrics Recommendation SHALL implement IPDR/SP Exporter functionality. For brevity, the term "metrics engine" is to be understood to include IPDR/SP Exporter functionality. The "Collector" is an implementation on the data receiving side of the IPDR/SP and is typically part of a back-office system.

The IPDR/SP utilizes a slightly augmented version of the XDR protocol, described in IETF RFC 1832, to encode IPDRs. IPDR/SP uses many of the primitive types defined by the IETF XDR protocol to encode as much information as possible in a compact binary representation.

The IPDR/SP uses TCP as its transport layer protocol. The transport layer acknowledgements are used to ensure the reliable delivery of data packets and detection of lost Exporters. In addition, IPDR/SP allows another level of application reliability as described below.

### **6.1.1 IPDR/SP Templates**

The IPDR/SP uses a highly-efficient encoding scheme, where only the values for each field of a particular type of IPDR are encapsulated and sent in a Data Message from the Exporter to the Collector. The field names, types, and lengths are not included in the IPDRs; instead, the field names, types, and lengths are specified indirectly by referring to a session-specific Template that describes contents of that type of IPDR.

Within the IPDR/SP, IPDRs are always exchanged within the context of a particular session that exists between a particular Exporter and a particular Collector. During the session establishment phase, the Exporter provides the Collector with a list of Templates, where each Template describes the layout of a particular type of Data Record (a Data Record is a binary encoding of an IPDR record) that can be sent to the Collector. Data Records are always packaged into a Data Message before being sent to the Collector.

Each Template definition contains a TemplateID, an IPDR-Type name, an explicit reference to the XML Schema which describes the IPDR-Type referred to by the Template, and an ordered list of triplets, where each triplet consists of a field name, field ID, and type. These field names and types must correspond to those defined in the corresponding IPDR Schema for the particular IPDR-Type referred to by the Template.

A particular Template not only specifies an IPDR-Type and corresponding XML Schema, but also specifies the particular subset of fields, and the order in which those fields will be laid out within Data Records referencing that Template. A Template for a particular IPDR-Type does not have to contain all possible fields defined in the XML Schema for that IPDR-Type; it might exclude certain optional or conditional fields. As such, there could be multiple Templates referring to the same IPDR-Type within a particular session.

Typically the Exporter sends a list of supported Templates to the Collector during the Session Establishment phase; however, the IPDR/SP also provides a mechanism whereby a Collector can negotiate with an Exporter regarding the Templates to be used during that session.

Once a session has been established, Data Messages are sent from the Exporter to the Collector. Each Data Message contains a sequence number, a TemplateID, which points to the particular Template that describes the contents of the Data Record contained within this particular Data Message, followed by a Data Record (a binary-encoded IPDR). An active session can simultaneously support many different IPDR-Types and many different subsets of fields for a particular IPDR-Type.

The use of Templates makes it possible for Exporters to deliver only the actual data values for each usage event without having to include any data descriptors in each Data Record. This significantly reduces the volume of information sent over communication links. The use of Templates also has the added advantage of enabling Data Records to contain only a pertinent subset of all fields for a particular IPDR-Type, thereby reducing the number of "empty" fields. This also reduces the volume of information sent over communication links.

## 6.1.2 System level reliability

During Session Establishment, the Exporter and the Collector agree on how often the Collector will send a Data Acknowledge Message back to the Exporter. Each Data Acknowledge Message refers to the sequence number of the most recently received and stored Data Message. By sending a Data Acknowledge Message, the Collector tells the Exporter that all Data Messages with sequence numbers less than or equal to the sequence number contained within the Data Acknowledge Message have been received and transferred to some type of persistent storage.

Once the Collector has acknowledged receipt of a particular Data Message, the Exporter no longer needs to maintain the Data Record contained within that Data Message. If the Collector goes down, or if the connection between the Exporter and Collector is disrupted, the Exporter can compare its most recently generated Data Message sequence number with the last acknowledged sequence number to know which Data Messages and Data Records it needs to preserve and either resend to the Collector or send to alternate Collectors. The Exporter can continue to stream new Data Messages to a Collector, regardless of whether previous Data Messages have been acknowledged, as long as the Collector sends intermittent Data Acknowledge Messages at the frequency agreed to at the start of the session.

A metrics engine SHALL adhere to the following requirements:

- The metrics engine SHALL NOT attempt to connect to an unauthorized Collector.
- The metrics engine SHALL NOT generate data elements unless connected to and configured to by an authorized Collector.

This Recommendation does not place requirements on Collectors. The following guidelines may be used by cable operators in order to support metrics collection.

- The Collectors SHOULD implement the IPDR/SP Collector functionality.
- One or more redundant Collectors SHOULD be configured such that there MAY be more than one Collector connected to one Exporter. This configuration provides high availability and improved robustness of the collection system.

## 6.2 Data Model

Within the IPDR/SP, every IPDR refers to its particular IPDR-Type (or type of event message), which then defines the contents that an IPDR of that IPDR-Type is expected to contain. IPDR/SP requires that all IPDR-Types be specified using the XML schema language. Defining event messages within IPDR/SP is a relatively easy and straightforward process.

[IPDR/SSDG] defines a human readable format for defining a generic data model, as an XML Schema. Annex A defines a data model for metrics using this format. Receivers that support metrics SHALL implement the data model defined in Annex A.

The XML schema defining a particular IPDR-Type SHALL contain the type-name for that IPDR-Type and a list of all possible fields that MAY appear within an IPDR of that type. Each field has a name, type, and an attribute specifying whether that field is required, optional, or conditional. Additionally, each field MAY be annotated with supplemental textual information, explaining the nature of the data contained in that field.

The IPDR/SP gives implementers nearly complete freedom to define the fields that are contained within a particular IPDR-Type.

XML schema provides an easy way for cable operators and vendors to add their own extensions to the base event message types defined in this Recommendation. Extensions MAY be added simply by creating a new XML schema, defining a new IPDR-Type name, importing fields from pre-existing "standard" IPDR-Types, and then adding definitions only for any new implementation-specific fields.

Since XML schema supports namespaces, there is a built-in mechanism for reducing the chances of field name collision, especially when using implementation-specific extensions, by placing implementation-specific fields into their own unique namespace. The "ipdr" namespace, which contains all of the IPDR defined fields, is managed by IPDR.org. Other entities are free to create their own namespaces, and place their fields within their own unique namespaces.

IPDR/SP defines a mechanism for transforming the XML human readable data model into a binary format optimized for on-the-wire transmission, via XDR. As part of their requirement to implement an IPDR/SP Exporter, receivers that support metrics SHALL be able to transmit a binary representation of the data model defined in Annex A.

Since XML schema is used to describe the contents of IPDRs, it is a simple exercise to translate an XDR encoded IPDR to an XML encoding, and vice versa. While it would be undesirable to send XML-encoded IPDRs over a high-traffic connection, once the IPDRs have arrived at a Collector there are circumstances where being able to translate those IPDRs from XDR to XML can be quite helpful for debugging, human readability, or translation to other encoding schemes.

## Annex A

### Metrics Data Model Schema

(This annex forms an integral part of this Recommendation)

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

<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:OCAP="http://www.ipdr.org/namespaces/CableLabs(R)/OCAP(TM)"
xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
targetNamespace="http://www.ipdr.org/namespaces/CableLabs(R)/OCAP(TM)" version="3.5-A.0"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="http://www.ipdr.org/namespaces/ipdr"
schemaLocation="http://www.ipdr.org/public/IPDRDoc3.5.1.xsd"/>
  <include schemaLocation="http://www.ipdr.org/public/IPDRTypes.xsd"/>

  <element name="protocolVersionMajor" type="byte">
</element>

  <element name="protocolVersionMinor" type="byte">
</element>

  <element name="hostID" type="string"/>

  <element name="hostType">
    <simpleType>
      <restriction base="int">
        <enumeration value="1">
          <annotation>
            <appinfo>
              <ipdr:enumMeaning>
                OCAP
              </ipdr:enumMeaning>
            </appinfo>
          </annotation>
        </enumeration>
        <enumeration value="2">
          <annotation>
            <appinfo>
              <ipdr:enumMeaning>
                ETV
              </ipdr:enumMeaning>
            </appinfo>
          </annotation>
        </enumeration>
        <enumeration value="3">
          <annotation>
            <appinfo>
              <ipdr:enumMeaning>
                other
              </ipdr:enumMeaning>
            </appinfo>
          </annotation>
        </enumeration>
      </restriction>
    </simpleType>
  </element>

  <element name="createTime" type="ipdr:dateTimeMsec">
</element>

  <element name="transmitTime" type="ipdr:dateTimeMsec">
</element>

  <element name="privateData" type="hexBinary">
</element>
```

```

<element name="serviceSelection">
  <complexType>
    <sequence>
      <element ref="createTime"/>
      <element name="sourceID" type="int">
      </element>
    </sequence>
  </complexType>
</element>

<element name="tuning">
  <complexType>
    <sequence>
      <element ref="createTime"/>
      <element name="tunerID" type="int">
      </element>
      <element name="frequency" type="int">
      </element>
      <element name="programNumber" type="int">
      </element>
    </sequence>
  </complexType>
</element>

<element name="SDV">
  <complexType>
    <sequence>
      <element ref="createTime"/>
      <element name="sourceID" type="int">
      </element>
    </sequence>
  </complexType>
</element>

<element name="servicePresentation">
  <complexType>
    <sequence>
      <element ref="createTime"/>
      <element name="sourceID" type="int">
      </element>
      <element name="aspectRatio">
        <simpleType>
          <restriction base="int">
            <enumeration value="1">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    4:3
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
            <enumeration value="2">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    16:9
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
          </restriction>
        </simpleType>
      </element>
      <element name="resolution" type="int">
      </element>
      <element name="audio">
        <simpleType>
          <restriction base="int">
            <enumeration value="0">

```

```

        <annotation>
            <appinfo>
                <ipdr:enumMeaning>
                    No audio
                </ipdr:enumMeaning>
            </appinfo>
        </annotation>
    </enumeration>
    <enumeration value="1">
        <annotation>
            <appinfo>
                <ipdr:enumMeaning>
                    Audio
                </ipdr:enumMeaning>
            </appinfo>
        </annotation>
    </enumeration>
</restriction>
</simpleType>
</element>
</sequence>
</complexType>
</element>

<element name="segmentPresentation">
    <complexType>
        <sequence>
            <element ref="createTime"/>
            <element name="segmentID" type="int">
            </element>
        </sequence>
    </complexType>
</element>

<element name="DPI">
    <complexType>
        <sequence>
            <element ref="createTime"/>
            <element name="sourceID" type="int">
            </element>
            <element name="replacementID" type="int">
            </element>
            <element name="replacementSource">
                <simpleType>
                    <restriction base="int">
                        <enumeration value="1">
                            <annotation>
                                <appinfo>
                                    <ipdr:enumMeaning>
                                        In transport
                                    </ipdr:enumMeaning>
                                </appinfo>
                            </annotation>
                        </enumeration>
                        <enumeration value="2">
                            <annotation>
                                <appinfo>
                                    <ipdr:enumMeaning>
                                        Out of transport
                                    </ipdr:enumMeaning>
                                </appinfo>
                            </annotation>
                        </enumeration>
                        <enumeration value="3">
                            <annotation>
                                <appinfo>
                                    <ipdr:enumMeaning>
                                        From Memory
                                    </ipdr:enumMeaning>
                                </appinfo>
                            </annotation>
                        </enumeration>
                    </restriction>
                </simpleType>
            </element>
        </sequence>
    </complexType>
</element>

```

```

        </enumeration>
    </restriction>
</simpleType>
</element>
<element name="switchLevel" type="int">
    <simpleType>
        <restriction base="int">
            <enumeration value="0">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            L0
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
            <enumeration value="1">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            L1
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
</element>
</sequence>
</complexType>
</element>

<element name="inputEvent">
    <complexType>
        <sequence>
            <element ref="createTime"/>
            <element name="keyCode" type="int">
            </element>
        </sequence>
    </complexType>
</element>

<element name="applicationLoadRequested">
    <complexType>
        <sequence>
            <element ref="createTime"/>
            <element name="appID" type="int">
            </element>
            <element name="appType">
                <simpleType>
                    <restriction base="int">
                        <enumeration value="1">
                            <annotation>
                                <appinfo>
                                    <ipdr:enumMeaning>
                                        OCAP
                                    </ipdr:enumMeaning>
                                </appinfo>
                            </annotation>
                        </enumeration>
                        <enumeration value="2">
                            <annotation>
                                <appinfo>
                                    <ipdr:enumMeaning>
                                        ETV
                                    </ipdr:enumMeaning>
                                </appinfo>
                            </annotation>
                        </enumeration>
                        <enumeration value="3">
                            <annotation>

```



```

        <appinfo>
          <ipdr:enumMeaning>
            other
          </ipdr:enumMeaning>
        </appinfo>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>
</element>
</sequence>
</complexType>
</element>

<element name="applicationLoaded">
  <complexType>
    <sequence>
      <element ref="createTime"/>
      <element name="appID" type="int">
      </element>
      <element name="appType">
        <simpleType>
          <restriction base="int">
            <enumeration value="1">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    OCAP
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
            <enumeration value="2">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    ETV
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
            <enumeration value="3">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    other
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
          </restriction>
        </simpleType>
      </element>
      <element name="numResourceFiles" type="int">
      </element>
      <element name="transportProtocol">
        <simpleType>
          <restriction base="int">
            <enumeration value="1">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    IBOC
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
            <enumeration value="2">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>

```

```

        OOBBC
        </ipdr:enumMeaning>
    </appinfo>
</annotation>
</enumeration>
<enumeration value="3">
    <annotation>
        <appinfo>
            <ipdr:enumMeaning>
                HTTP
            </ipdr:enumMeaning>
        </appinfo>
    </annotation>
</enumeration>
</restriction>
</simpleType>
</element>
<element name="controlCode">
    <simpleType>
        <restriction base="int">
            <enumeration value="1">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            LOADED
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
            <enumeration value="2">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            PAUSED
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
            <enumeration value="3">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            RUNNING
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
</element>
</sequence>
</complexType>
</element>
<element name="applicationLifecycle">
    <complexType>
        <sequence>
            <element ref=" createTime"/>
            <element name="appID" type="int"/>
            <element name="appType">
                <simpleType>
                    <restriction base="int">
                        <enumeration value="1">
                            <annotation>
                                <appinfo>
                                    <ipdr:enumMeaning>
                                        OCAP
                                    </ipdr:enumMeaning>
                                </appinfo>
                            </annotation>
                        </enumeration>
                    </restriction>
                </simpleType>
            </element>
        </sequence>
    </complexType>
</element>

```

```

        <enumeration value="2">
            <annotation>
                <appinfo>
                    <ipdr:enumMeaning>
                        ETV
                    </ipdr:enumMeaning>
                </appinfo>
            </annotation>
        </enumeration>
        <enumeration value="3">
            <annotation>
                <appinfo>
                    <ipdr:enumMeaning>
                        other
                    </ipdr:enumMeaning>
                </appinfo>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>
</element>
<element name="lifecycleState">
    <simpleType>
        <restriction base="int">
            <enumeration value="1">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            LAUNCHED
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
            <enumeration value="2">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            PAUSED
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
            <enumeration value="3">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            RESUMED
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
            <enumeration value="4">
                <annotation>
                    <appinfo>
                        <ipdr:enumMeaning>
                            TERMINATED
                        </ipdr:enumMeaning>
                    </appinfo>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
</element>
</sequence>
</complexType>
</element>

<element name="environmentSelection">
    <complexType>
        <sequence>
            <element ref="createTime"/>

```

```

<element name="environment">
  <simpleType>
    <restriction base="int">
      <enumeration value="1">
        <annotation>
          <appinfo>
            <ipdr:enumMeaning>
              Cable
            </ipdr:enumMeaning>
          </appinfo>
        </annotation>
      </enumeration>
      <enumeration value="2">
        <annotation>
          <appinfo>
            <ipdr:enumMeaning>
              Non cable
            </ipdr:enumMeaning>
          </appinfo>
        </annotation>
      </enumeration>
    </restriction>
  </simpleType>
</element>
</sequence>
</complexType>
</element>

<element name="applicationMode">
  <complexType>
    <sequence>
      <element ref=" createTime"/>
      <element name="appID" type="int"/>
      <element name="appType"/>
      <element name="mode">
        <simpleType>
          <restriction base="int">
            <enumeration value="1">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    CROSS_ENVIRONMENT
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
            <enumeration value="2">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    BACKGROUND
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
            <enumeration value="3">
              <annotation>
                <appinfo>
                  <ipdr:enumMeaning>
                    NORMAL
                  </ipdr:enumMeaning>
                </appinfo>
              </annotation>
            </enumeration>
          </restriction>
        </simpleType>
      </element>
    </sequence>
  </complexType>
</element>

```

```

<element name="application">
  <complexType>
    <sequence>
      <element ref="createTime"/>
      <element name="privateData" type="hexBinary">
        </element>
      </sequence>
    </complexType>
  </element>

<complexType name="OCAP-Type">
  <complexContent>
    <extension base="ipdr:IPDRType">
      <sequence>
        <element ref="protocolVersionMajor"/>
        <element ref="protocolVersionMinor"/>
        <element ref="hostID"/>
        <element ref="hostType"/>
        <element ref="createTime"/>
        <element ref="transmitTime"/>
        <element ref="privateData"/>
        <element ref="serviceSelection" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="tuning" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="SDV" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="servicePresentation" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="segmentPresentation" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="DPI" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="inputEvent" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="applicationLoadRequested" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="applicationLoaded" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="applicationLifecycle" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="environmentSelection" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="applicationMode" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="application" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

</schema>

```

## Annex B

### Schema Data Element Requirements

(This annex forms an integral part of this Recommendation)

This Recommendation may be referenced by [ETV] or other platforms.

Where semantics for the data elements of Annex A may be consistent across all platforms, they are defined in this annex.

Where semantics for the data elements of Annex A are platform specific, a platform definition that includes this data model specification is expected to further define the element. In such cases, Platform Specific Semantics will further define an element.

#### **B.1 protocolVersionMajor**

This 8-bit field SHALL indicate the major version of this Recommendation. This field SHALL have the value 1.

#### **B.2 protocolVersionMinor**

This 8-bit field SHALL indicate the minor version of this Recommendation. This field SHALL have the value 0.

#### **B.3 hostID**

This string field SHALL indicate a unique host identifier. This value SHALL be defined by Platform Specific Semantics.

#### **B.4 hostType**

This enumerated integer field SHALL indicate the Host type. This value SHALL be one of: 1 for OCAP, 2 for ETV, or 3 for other, as selected by Platform Specific Semantics.

#### **B.5 createTime**

This dateTimeMsec field SHALL indicate when the metrics log was created or last flushed by the platform.

#### **B.6 transmitTime**

This dateTimeMsec field SHALL indicate when the metrics log was last transmitted to the headend by the platform.

#### **B.7 privateData**

This hexBinary field MAY be generated by the platform for private use.

#### **B.8 serviceSelection**

This complex field SHALL indicate the time when a new service has been selected, together with the integer Source ID of that new service.

The *sourceID* value SHALL be defined by Platform Specific Semantics.

## **B.9 tuning**

This complex field SHALL indicate the time when an RF tuning action has been performed, together with the integer number of the tuner that performed the tuning operation, the integer frequency of the MPEG-2 transport stream tuned to, and the integer program number of the MPEG-2 transport stream tuned to.

The *tunerID* value SHALL be defined by Platform Specific Semantics.

The *frequency* value SHALL be equal to the RF frequency acquired by the receiver.

The *programNumber* value SHALL be equal to the MPEG-2 Program Number acquired by the receiver.

## **B.10 SDV**

This complex field SHALL indicate the time when a new Switched Digital Service has been selected, and the integer Source ID of that new service.

The *sourceID* value SHALL be defined by Platform Specific Semantics.

## **B.11 servicePresentation**

This complex field SHALL indicate the time when presentation of a new service has started, together with the integer Source ID of the service being presented, the enumerated integer aspect ratio of the display area of the presentation, the integer resolution of the display area of the presentation, and the enumerated integer indication of whether audio is on or off for the service being presented.

The *sourceID* value SHALL be defined by Platform Specific Semantics.

The *aspectRatio* value SHALL be one of: 1 for 4:3, 2 for 16:9.

The *resolution* value SHALL be defined by Platform Specific Semantics.

The *audio* value SHALL be one of: 0 for audio off, 1 for audio on.

## **B.12 segmentPresentation**

This complex field SHALL indicate the time when a new segmentation descriptor is detected within a presentation by the platform, together with the integer Segment ID of that new segment as taken from that descriptor. Presentations MAY be from broadcast, from a DVR recording, from SDV, or from an On-Demand session.

The *segmentID* value SHALL be defined by Platform Specific Semantics.

## **B.13 DPI**

This complex field SHALL indicate the time when an existing service has been replaced by new Digital Program Insertion service, together with the integer Source ID of the service being replaced, the integer Source ID of the new destination service, the enumerated integer source of the replacement, and the integer switch level of the replacement.

The *sourceID* and *replacementID* values SHALL be defined by Platform Specific Semantics.

The *replacementSource* value SHALL be one of: 1 for In-transport, 2 for Out-of-transport, 3 for From-Memory.

The *switchLevel* value SHALL be one of: 0 for L0 switch, 1 for L1 switch.

#### **B.14 inputEvent**

This complex field SHALL indicate the time when a key click from a remote control or other input device has been detected by the platform, together with the integer *KeyCode* value itself.

The *keyCode* value SHALL be defined by Platform Specific Semantics.

#### **B.15 applicationLoadRequested**

This complex field SHALL indicate the time when the platform has received a request to load an application, together with an integer value that identifies the application and an enumerated integer application type.

The *appID* value SHALL be defined by Platform Specific Semantics.

The *appType* value SHALL be one of: 1 for OCAP, 2 for ETV, or 3 for other, as selected by Platform Specific Semantics.

#### **B.16 applicationLoaded**

This complex field SHALL indicate the time when the platform has successfully loaded an application, together with an integer value that identifies the application, an enumerated integer application type, an integer value that specifies the number of resource files that have currently been loaded for the application, the enumerated integer transport load mechanism, and the enumerated initial state of the application.

The *appID* value SHALL be defined by Platform Specific Semantics.

The *appType* value SHALL be one of: 1 for OCAP, 2 for ETV, or 3 for other, as selected by Platform Specific Semantics.

The *numResourceFiles* value SHALL be defined by Platform Specific Semantics.

The *transportProtocol* value SHALL be one of: 1 for In-Band-Object-Carousel, 2 for Out-Of-Band-Object-Carousel, 3 for HTTP, as selected by Platform Specific Semantics.

The *controlCode* value SHALL be one of: 1 for LOADED, 2 for PAUSED, 3 for RUNNING, as selected by Platform Specific Semantics.

#### **B.17 applicationLifecycle**

This complex field SHALL indicate the time when the platform has modified the state of an application, together with an integer value that identifies the application and an enumerated integer application type.

The *appID* value SHALL be defined by Platform Specific Semantics.

The *appType* value SHALL be one of: 1 for OCAP, 2 for ETV, or 3 for other, as selected by Platform Specific Semantics.

The *lifeCycleState* value SHALL be one of: 1 for LAUNCHED, 2 for PAUSED, 3 for RESUMED, 4 for TERMINATED, as selected by Platform Specific Semantics.

#### **B.18 environmentSelection**

This complex field SHALL indicate the time when the platform has successfully switched to a new operating environment, together with the enumerated integer environment identifier.

The *environment* value SHALL be one of: 1 for Cable, 2 for Non-Cable, as selected by Platform Specific Semantics.



### **B.19 applicationMode**

This complex field SHALL indicate the time when the platform has modified the mode of an application, together with an integer value that identifies the application and an enumerated integer application type.

The *appID* value SHALL be defined by Platform Specific Semantics.

The *appType* value SHALL be one of: 1 for OCAP, 2 for ETV, or 3 for other, as selected by Platform Specific Semantics.

The *mode* value SHALL be one of: 1 for CROSS-ENVIRONMENT, 2 for BACKGROUND, 3 for NORMAL, as selected by Platform Specific Semantics.

### **B.20 application**

This complex field SHALL indicate the time when an application has conveyed a private data set to the platform, together with the binary data itself. The hexBinary *privateData* value SHALL be forwarded to the metrics collection without further interpretation.





## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
<b>Series J</b>	<b>Cable networks and transmission of television, sound programme and other multimedia signals</b>
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems