INTERNATIONAL STANDARD

**ISO/IEC 23001-3**

First edition
2008-10-01

# Information technology — MPEG systems technologies —

## Part 3: XML IPMP messages

*Technologies de l'information — Technologies des systèmes MPEG —*

*Partie 3: Messages XML IPMP*

Reference number
ISO/IEC 23001-3:2008(E)

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

**COPYRIGHT PROTECTED DOCUMENT**

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

## Contents

Page

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 23001-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23001 consists of the following parts, under the general title *Information technology — MPEG systems technologies*:

— *Part 1: Binary MPEG format for XML*

— *Part 2: Fragment request units*

— *Part 3: XML IPMP messages*

— *Part 4: Codec configuration representation*

— *Part 5: Bitstream Syntax Description Language (BSDL)*

# Introduction

This part of ISO/IEC 23001 specifies a set of XML messages exchanged between the components of a device or between devices while performing Intellectual Property Management and Protection (IPMP) operations.

The protection mechanisms are based on the IPMP Tool model as defined in ISO/IEC 13818-11, ISO/IEC 14496-13 (MPEG-2/4 IPMP Extensions), and ISO/IEC 21000-4 (MPEG-21 IPMP Components). In order to support the operation of IPMP Tools on a device in an interoperable fashion, this part of ISO/IEC 23001 complements ISO/IEC 21000-4 by defining an XML API for the communication between IPMP Tools and the device on which they operate, or between two IPMP Tools.

# Information technology — MPEG systems technologies —

# Part 3: XML IPMP messages

## 1 Scope

This part of ISO/IEC 23001 specifies XML IPMP messages (hereinafter IPMP Messages) which are a simple and natural extension of the IPMP Information Descriptors defined in ISO/IEC 21000-4. They allow dispatching of the IPMP information related to a protected content element retrieved from the associated digital item to the modules in charge of performing the IPMP operations required to access the protected content element. Furthermore, the XML IPMP Messages provide a standard API allowing the exchange of IPMP information (e.g. decryption keys, authentication information, licenses, etc.) between the modules – the IPMP Tools – in charge of providing access to the protected content elements (e.g. audio/visual resources).

The API specified by this part of ISO/IEC 23001 is based on a set of XML messages defining the format of the information exchanged between the various components on a device, or between two devices. Most of the IPMP Messages defined in this specification are a translation of the messages originally defined in ISO/IEC 13818-11 and ISO/IEC 14496-13 from the binary representation into an XML representation. A number of these XML messages have been adapted in order to harmonise ISO/IEC 13818-11 and ISO/IEC 14496-13 in the MPEG-21 Multimedia Framework and with ISO/IEC 21000-4 specifically. Additionally, a number of new XML messages have been defined to support extended IPMP requirements such as the separation between the IPMP algorithms and the logic enabling them to communicate with the other components in a device.

The XML IPMP Messages are employed in two different contexts:

a. Intra-device communication: enabling interoperable communication among two IPMP Tools or the IPMP Processor and IPMP Tools (allowing for instance the two to authenticate or to exchange IPMP information), as well as performing management tasks of different components on a device.

b. Inter-device communication: in a context where digital items are streamed from one device to another by means of the Digital Item Streaming technology (addressed by ISO/IEC 23000-5), a standard set of IPMP Messages extending the IPMP Information Descriptor enables interoperable communication and management of IPMP Tools running on a device from a remote device streaming the digital item.

The XML IPMP messages defined in this part of ISO/IEC 23001 are grouped in the following macro-categories:

- IPMP Message Containers: the containers for IPMP Messages
- Mutual Authentication Messages: messages enabling one module to request mutual authentication with another one and to carry out the necessary steps in order to achieve it
- IPMP Tool Connection and Disconnection messages: enabling one IPMP Tool to request the instantiation as well as the disposal of another IPMP Tool
- IPMP Tool Notification Messages: enabling one component to request notification in case certain events occur, and conveying information about the event occurred
- IPMP Processing Messages: a set of messages enabling the exchange of IPMP information ranging from licenses and decryption keys, watermarking data, etc.
- User Interaction Messages: allowing the exchange of information between an IPMP Tool and a user or vice versa

- Additional IPMP Messages: messages that on the one hand allow a greater decoupling between the IPMP algorithm implementations and the logic necessary to instantiate/initialize/manage them, and on the other do not require content providers and IPMP Tool providers to disclose the details of the IPMP modules
- Legacy Messages: messages originally defined in ISO/IEC 13818-11 and ISO/IEC 14496-13 and translated to XML for the sake of completeness.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 13818-11, *Information technology — Generic coding of moving pictures and associated audio information — Part 11: IPMP on MPEG-2 systems*

ISO/IEC 14496-13:2004, *Information technology — Coding of audio-visual objects — Part 13: Intellectual Property Management and Protection (IPMP) extensions*

ISO/IEC 21000-4:2006, *Information technology — Multimedia framework (MPEG-21) — Part 4: Intellectual Property Management and Protection Components*

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

**3.1**
**device**
combination of hardware and software or just an instance of software that allows a user to perform actions

**3.2**
**IPMP Processor**
conceptual entity within a Media Streaming Player that executes Digital Rights Management (DRM)-related functions on **IPMP Tools** such as instantiating, authenticating and managing them during their life cycle

**3.3**
**IPMP Tool**
module performing (one or more) IPMP functions such as authentication, decryption, watermarking, etc.

**3.4**
**IPMP Tool Agent**
module instantiating, initialising, authenticating, and supervising any operation performed between **IPMP Tools** within an **IPMP Tool Group**

**3.5**
**IPMP Tool Body**
executable code implementing either a **Single IPMP Tool** or an **IPMP Tool Pack**

**3.6**
**IPMP Tool Group**
combination of several **IPMP Tools**

**3.7**
**IPMP Tool Pack**
module that comprises an **IPMP Tool Group** and its **IPMP Tool Agent**

**3.8**
**IPMP Tool Provider Device**
**device** delivering **IPMP Tools** to another **device**

**3.9**
**Resource Processor**
conceptual entity responsible for handling resources in a **device**

**3.10**
**user**
any identifiable person or legal entity using a **device**

NOTE Some terms and definitions listed in this section were originally defined in ISO/IEC 13818-11 and ISO/IEC 14496-13 with a pure MPEG-2/MPEG-4 scope. As this part of ISO/IEC 23001 specifies technologies that may be applied to a broader MPEG-21 context, as specified in ISO/IEC 21000-4, which may include the former two as a specialization of some of its parts, some terms have been re-defined with the purpose of broadening their scope.

## 4 Abbreviated terms

List of abbreviated terms.

| | |
|---|---|
| DID | Digital Item Declaration |
| DIDL | Digital Item Declaration Language |
| DII | Digital Item Identification |
| IPMP | Intellectual Property Management and Protection |
| URI | Uniform Resource Identifier |

## 5 Namespace conventions

Throughout this part of ISO/IEC 23001, Qualified Names are written with a namespace prefix followed by a colon followed by the local part of the Qualified Name.

For clarity, throughout this part of ISO/IEC 23001, consistent namespace prefixes are used. Table 1 gives these prefixes and the corresponding namespace.

**Table 1 — List of namespaces and corresponding prefixes defined in/referenced by this specification**

| Prefix | Corresponding namespace |
|---|---|
| ipmpdidl | urn:mpeg:mpeg21:2004:01-IPMPDIDL-NS |
| ipmpmsg | urn:mpeg:mpegB:schema:XML-IPMP-MESSAGES:2007 |
| ipmpinfo | urn:mpeg:mpeg21:2004:01-IPMPINFO-NS |
| didl | urn:mpeg:mpeg21:2002:02-DIDL-NS |
| didmodel | urn:mpeg:mpeg21:2002:02-DIDMODEL-NS |
| dii | urn:mpeg:mpeg21:2002:01-DII-NS |
| r | urn:mpeg:mpeg21:2003:01-REL-R-NS |
| xsd | http://www.w3.org/2001/XMLSchema |
| xsi | http://www.w3.org/2001/XMLSchema-instance |
| dsig | http://www.w3.org/2000/09/xmldsig# |
| xenc | http://www.w3.org/2001/04/xmlenc# |

# 6 IPMP Messages Specification

## 6.1 Introduction

This clause provides the syntax and the semantics of the IPMP Messages expressed in the XML Schema Definition notation.

There are two types of IPMP Messages, specified in the sub-sections below:

- IPMP Message Containers
- IPMP Messages

IPMP Messages are defined in the namespace characterised by the following URI: "`urn:mpeg:mpegB:schema:XML-IPMP-MESSAGES:2007`", and identified by the prefix "ipmpmsg".

Note In the sub-sections below an IPMP Tool can be either a single IPMP Tool or an IPMP Tool Agent of an IPMP Tool Pack.

## 6.2 IPMP Message Containers

### 6.2.1 Introduction

An IPMP Processor instantiates an IPMP Tool in order to access a protected resource represented by an IPMPDIDL element, every time it encounters an ipmpinfo:Tool element within an IPMPDIDL element.

The IPMP Processor assigns an identifier to each instance of IPMP Tool, named the IPMP Tool Context ID. The Tool Context ID serves a different role from the IPMP Tool ID. In particular, two instances of an IPMP Tool with a common IPMP Tool ID will be assigned different values of Tool Context ID.

Once the IPMP Tool is instantiated, the access of the governed content element through the IPMP Tool may involve the exchange of information between the IPMP Processor and the IPMP Tool or between the various IPMP Tools. This information is encapsulated within XML IPMP Messages, which are encapsulated in two different containers depending on the way the message is originated:

a. MessageFromDID: a container message sent by the IPMP Processor to an IPMP Tool conveying information extracted from the digital item and addressed to the IPMP Tool (e.g. a decryption key necessary for decrypt a resource);
b. MessageFromTool: a container message sent either from the IPMP Processor or from an IPMP Tool to another IPMP Tool, containing information originated internally by the sender, e.g. a message requesting mutual authentication.

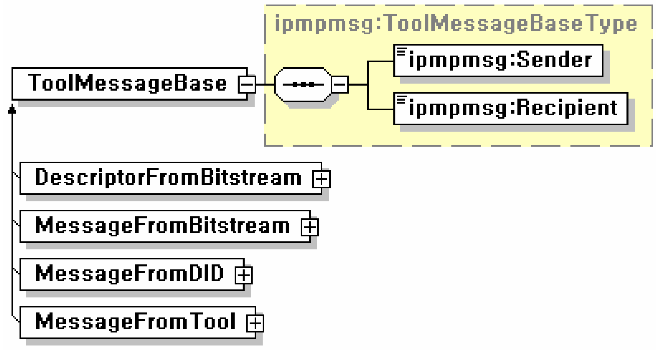
### 6.2.2 IPMPBaseType

Both containers messages, MessageFromDID and MessageFromTool defined above, extend the same abstract XML element Base Type: ipmpmsg:IPMPBaseType defined as following:

```
<complexType name="IPMPBaseType" abstract="true"/>
```

### 6.2.3 ToolMessageBase

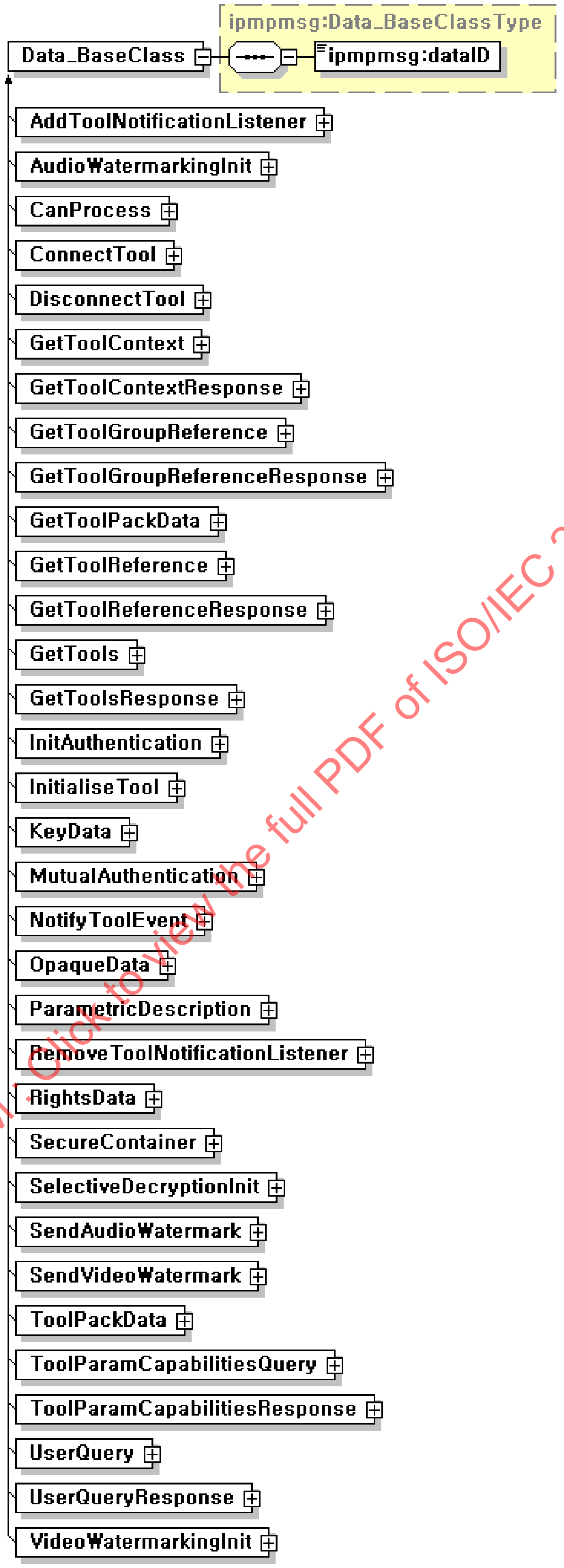
All IPMP Message Containers are based on the ipmpmsg:ToolMessageBase element.

| | |
|---|---|
| **Diagram** |   ipmpmsg:ToolMessageBaseType<br>ToolMessageBase<br>ipmpmsg:Sender<br>ipmpmsg:Recipient<br>DescriptorFromBitstream<br>MessageFromBitstream<br>MessageFromDID<br>MessageFromTool |
| **Children** | `<Sender>` `<Recipient>` |
| **Source** | `<element name="ToolMessageBase" type="ipmpmsg:ToolMessageBaseType" abstract="true"/>`<br>`<complexType name="ToolMessageBaseType" abstract="true">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:IPMPBaseType">`<br>`            <sequence>`<br>`                <element name="Sender" type="unsignedInt"/>`<br>`                <element name="Recipient" type="unsignedInt"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

The abstract element ipmpmsg:ToolMessageBase extends ipmpmsg:IPMPBaseType by adding Sender and Recipient elements to the message. These elements contain an unsigned integer representing the unique Identifier that the IPMP terminal assigns to each instance of an IPMP Tool, the IPMP Tool Context ID. The IPMP terminal shall be identified by the value ‘0’.

### 6.2.4 Data_BaseClass

All IPMP Messages extend the ipmpmsg:Data_BaseClass element.

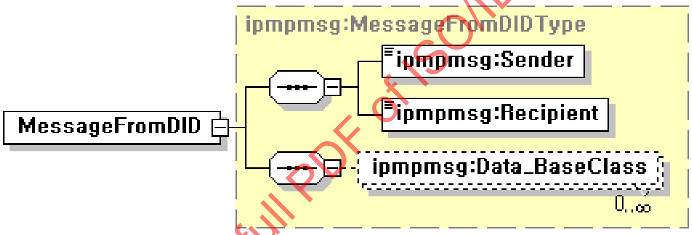
| | |
|---|---|
| **Diagram** |  |
| **Children** | <dataID> |
| **Source** | <element name="Data_BaseClass" type="ipmpmsg:Data_BaseClassType"<br>abstract="true"/><br><complexType name="Data_BaseClassType" abstract="true"> |

| | |
|---|---|
| |     &lt;annotation&gt;<br>        &lt;documentation&gt;alternates&lt;/documentation&gt;<br>    &lt;/annotation&gt;<br>    &lt;complexContent&gt;<br>        &lt;extension base="ipmpmsg:IPMPBaseType"&gt;<br>            &lt;sequence&gt;<br>                &lt;element name="dataID" type="unsignedInt"/&gt;<br>            &lt;/sequence&gt;<br>        &lt;/extension&gt;<br>    &lt;/complexContent&gt;<br>&lt;/complexType&gt; |

The abstract element ipmpmsg:Data_BaseClass extends the ipmpmsg:IPMPBaseType element by adding the dataID element, which shall contain an unsigned integer value identifying a message. The same value of dataID shall be used in any reply to that message.

### 6.2.5 MessageFromDID

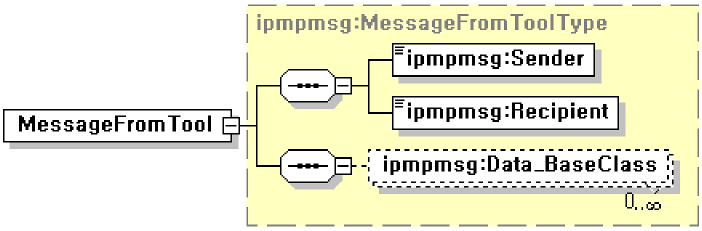
Any IPMP Message extracted from the digital item during its parsing by the IPMP terminal and addressed to a IPMP Tool shall be conveyed to that IPMP Tool by encapsulation in a ipmpmsg:MessageFromDID message.

| | |
|---|---|
| **Diagram** |  MessageFromDID<br>ipmpmsg:MessageFromDIDType<br>ipmpmsg:Sender<br>ipmpmsg:Recipient<br>ipmpmsg:Data_BaseClass<br>0..∞ |
| **Children** | &lt;Sender&gt; &lt;Recipient&gt; &lt;Data_BaseClass&gt; |
| **Source** | &lt;element name="MessageFromDID" type="ipmpmsg:MessageFromDIDType"<br>substitutionGroup="ipmpmsg:ToolMessageBase"/&gt;<br>    &lt;complexType name="MessageFromDIDType"&gt;<br>        &lt;complexContent&gt;<br>            &lt;extension base="ipmpmsg:ToolMessageBaseType"&gt;<br>                &lt;sequence&gt;<br>                    &lt;element ref="ipmpmsg:Data_BaseClass" minOccurs="0"<br>maxOccurs="unbounded"/&gt;<br>                &lt;/sequence&gt;<br>            &lt;/extension&gt;<br>        &lt;/complexContent&gt;<br>    &lt;/complexType&gt; |

The ipmpmsg:MessageFromDID element extends the ipmpmsg:ToolMessageBaseType complex type by adding a sequence of IPMP Messages extending the ipmpmsg:Data_BaseClass.

### 6.2.6 MessageFromTool

Any IPMP Message originated spontaneously from a IPMP Processor or an IPMP Tool and addressed to another IPMP Tool shall be conveyed to that IPMP Tool by encapsulation in a ipmpmsg:MessageFromTool message.

| | |
|---|---|
| **Diagram** |   ipmpmsg:MessageFromToolType<br>ipmpmsg:Sender<br>ipmpmsg:Recipient<br>MessageFromTool<br>ipmpmsg:Data_BaseClass<br>0..∞ |
| **Children** | <Sender> <Recipient> <Data_BaseClass> |
| **Source** | <element name="MessageFromTool" type="ipmpmsg:MessageFromToolType"<br>substitutionGroup="ipmpmsg:ToolMessageBase"/><br>    <complexType name="MessageFromToolType"><br>        <complexContent><br>            <extension base="ipmpmsg:ToolMessageBaseType"><br>                <sequence><br>                    <element ref="ipmpmsg:Data_BaseClass" minOccurs="0"<br>maxOccurs="unbounded"/><br>                </sequence><br>            </extension><br>        </complexContent><br>    </complexType> |

The ipmpmsg:MessageFromTool element (indirectly) extends the ipmpmsg:ToolMessageBase element by adding a sequence of IPMP Messages extending the ipmpmsg:Data_BaseClass.

## 6.3 IPMP Messages

### 6.3.1 Introduction

IPMP Messages express in an interoperable way a vast quantity of IPMP-related information which can be exchanged by two parties.

The delivery of IPMP Messages exchanged between two IPMP Tools on a device shall be handled by the IPMP Processor. Hence, a compliant IPMP Processor shall be able to forward all IPMP Messages defined in this specification to a recipient IPMP Tool. However, a compliant IPMP Processor is not required to be able to process all IPMP Messages, as some of them may contain information that is only meaningful for a specific IPMP Tool. A compliant IPMP Processor shall be able to process all IPMP Messages labelled “**IP**” in the Table below.

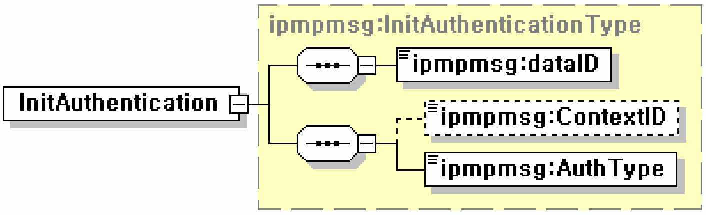
IPMP Messages are divided into two categories: those belonging to a base and those belonging to an extended profile of this specification. All IPMP Tools shall be able to process IPMP Messages belonging to the base profile, labelled with “**B**” in the table below. IPMP Tools may also be able to process IPMP Messages belonging to the extended profile, labelled with “**X**” in the table below, in which case they will be compliant with the extended profile.

### 6.3.2 Mutual Authentication Messages

The following table specifies the IPMP Messages to be exchanged between two IPMP Tools or between an IPMP Tool and the IPMP Processor to achieve Mutual Authentication.

#### 6.3.2.1 InitAuthentication

The InitAuthentication Message shall be employed by an entity to initialise the mutual authentication process with another entity.

| | |
|---|---|
| **Diagram** |  ipmpmsg:InitAuthenticationType<br>InitAuthentication<br>ipmpmsg:dataID<br>ipmpmsg:ContextID<br>ipmpmsg:AuthType |
| **Children** | `<dataID> <ContextID> <AuthType>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="InitAuthentication" type="ipmpmsg:InitAuthenticationType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="InitAuthenticationType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="ContextID" type="anyURI" minOccurs="0"/>`<br>`                <element name="AuthType" type="ipmpmsg:AUTType"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>`<br>`<simpleType name="AUTType">`<br>`    <restriction base="integer">`<br>`        <enumeration value="01"/>`<br>`        <enumeration value="02"/>`<br>`        <enumeration value="03"/>`<br>`        <enumeration value="04"/>`<br>`    </restriction>`<br>`</simpleType>` |

The InitAuthentication message extends the Data_BaseClass type by conveying:

- ContextID: the local ID of the IPMP Tool with which mutually authentication is to be initialised. This element is used when mutual authentication involves two IPMP Tools or an IPMP Processor and an IPMP Tool, or any time the sender knows the URI of the entity it is requesting to authenticate with.
- AuthType: the type of authentication required, which can only be given one of the values listed in the Table below, with the semantics specified in the right column.

**Table 2 — List of available Authentication Request types**

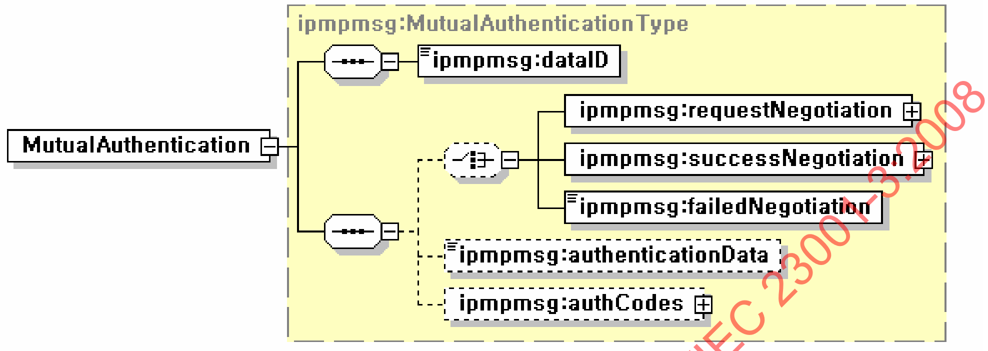
| | |
|---|---|
| 01 | No Authentication Required |
| 02 | No ID verify, Do secure channel |
| 03 | Do ID verify, No secure channel |
| 04 | Do ID verify, Do secure channel |

The recipient of an InitAuthentication messages shall reply with a MutualAuthentication message as defined in the sub-section below.

### 6.3.2.2 MutualAuthentication

The MutualAuthentication is employed by two entities (e.g. the IPMP Processor and an IPMP Tool) for the purpose of:

a. negotiating the authentication protocol
b. carrying out the agreed upon protocol
c. negotiating how the secured communication channel has to be used

| | |
|---|---|
| **Diagram** |   ipmpmsg:MutualAuthenticationType<br>MutualAuthentication<br>ipmpmsg:dataID<br>ipmpmsg:requestNegotiation<br>ipmpmsg:successNegotiation<br>ipmpmsg:failedNegotiation<br>ipmpmsg:authenticationData<br>ipmpmsg:authCodes |
| **Children** | `<dataID> <ContextID> <AuthType>` |
| **IP** | **IP** |
| **Profile** | **B** |
| **Source** | <element name="MutualAuthentication" type="ipmpmsg:MutualAuthenticationType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="MutualAuthenticationType"><br><complexContent><br><extension base="ipmpmsg:Data_BaseClassType"><br><sequence><br><choice minOccurs="0"><br><element name="requestNegotiation"<br>type="ipmpmsg:requestNegotiationType"/><br><element name="successNegotiation"<br>type="ipmpmsg:successNegotiationType"/><br><element name="failedNegotiation" type="boolean"<br>fixed="true"/><br></choice><br><element name="authenticationData" type="hexBinary"<br>minOccurs="0"/><br><element name="authCodes" type="ipmpmsg:AuthCodesType"<br>minOccurs="0"/><br></sequence><br></extension><br></complexContent><br></complexType> |

The MutualAuthentication message extends the Data_BaseClass Type by conveying:

- the choice between:
    - requestNegotiation element, when the message is employed in the initial phase of the authentication protocol in reply to an InitAuthentication message, to indicate that the sender is requesting negotiation about an authentication algorithm to be used in the subsequent communication
    - successNegotiation element, when the message is employed to indicate that the negotiation of an authentication protocol was successful, and for specifying the selected protocol.
    - failedNegotiation, indicating that none of the proposed authentication algorithms are supported.

- authenticationData, an optional element specifying data to be used for mutual authentication and whose value depends on method specific processing
- authCodes, optional authentication codes to this message.

The syntax of the requestNegotiationType complex type is defined below:

```
<complexType name="requestNegotiationType">
    <sequence>
        <element name="candidateAlgorithms" type="ipmpmsg:AlgorithmDescriptorType"/>
    </sequence>
</complexType>
```

The requestNegotiationType complex type conveys the list of authentication algorithms supported by means of the candidateAlgorithms element. The AlgorithmDescriptorType complex type is defined below.

```
<complexType name="AlgorithmDescriptorType">
    <sequence>
        <element name="algoID" type="anyURI" maxOccurs="unbounded"/>
        <element name="opaqueData" type="base64Binary" minOccurs="0"/>
    </sequence>
</complexType>
```

Each of the supported algorithms is characterised by an identifier of type xsd:anyURI. Optionally, the opaqueData element containing data related to the algorithm may be conveyed too.

The syntax of the successNegotiationType complex type is defined below:

```
<complexType name="successNegotiationType">
    <sequence>
        <element name="agreedAlgorithms" type="ipmpmsg:AlgorithmDescriptorType"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
```

The MutualAuthentication message containing the successNegotiation element is sent by the entity initiating the mutual authentication process in reply to a MutualAuthentication message proposing a list of candidate algorithms. The agreedAlgorithms element conveys the list of the Authentication algorithms supported, among the ones proposed by the entity that was challenged.

The syntax of the AuthCodesType complex type is defined below:

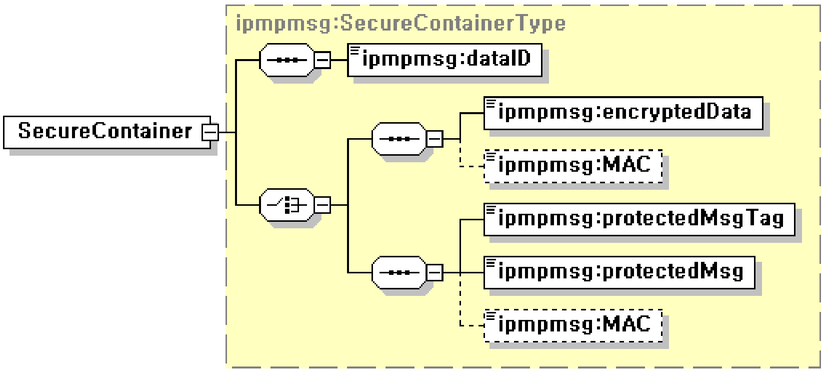
```
<complexType name="AuthCodesType">
    <sequence>
        <element name="certificates" type="dsig:KeyInfoType" maxOccurs="unbounded"/>
        <element name="trustData" type="hexBinary" minOccurs="0"/>
    </sequence>
</complexType>
```

The authCodes element conveys

- a number of certificates and/or trust data belonging to the entity involved in the Authentication process
- (optional) trust and security data, mostly used when Mutual Authentication involves IPMP Tools.

#### 6.3.2.3 SecureContainer

This message provides a secure container for any message extending IPMP_Data_BaseClass.

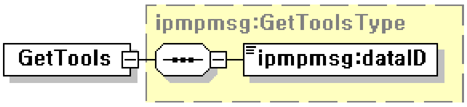
| | |
|---|---|
| **Diagram** |  ipmpmsg:SecureContainerType<br>SecureContainer<br>ipmpmsg:dataID<br>ipmpmsg:encryptedData<br>ipmpmsg:MAC<br>ipmpmsg:protectedMsgTag<br>ipmpmsg:protectedMsg<br>ipmpmsg:MAC |
| **Children** | `<dataID> <encryptedData> <MAC> <protectedMsgTag> <protectedMsg>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="SecureContainer" type="ipmpmsg:SecureContainerType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="SecureContainerType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <choice>`<br>`                <sequence>`<br>`                    <element name="encryptedData" type="hexBinary"/>`<br>`                    <element name="MAC" type="hexBinary" minOccurs="0"/>`<br>`                </sequence>`<br>`                <sequence>`<br>`                    <element name="protectedMsgTag" type="short"/>`<br>`                    <element name="protectedMsg" type="hexBinary"/>`<br>`                    <element name="MAC" type="hexBinary" minOccurs="0"/>`<br>`                </sequence>`<br>`            </choice>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

### 6.3.3 IPMP Tool Connection and Disconnection Messages

The following table specifies the IPMP Messages to be exchanged between IPMP Tools and the IPMP Processor to trigger the instantiation of a new IPMP Tool, or to obtain specific information about IPMP Tools.

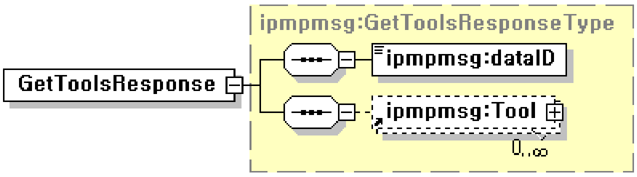
#### 6.3.3.1 GetTools

Message sent by an IPMP Tool to the IPMP Processor to find all the tools, instantiated or not, that are available on the device.

| | |
|---|---|
| **Diagram** |  |
| **Children** | `<dataID>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetTools" type="ipmpmsg:GetToolsType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolsType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType"/>`<br>`    </complexContent>`<br>`</complexType>` |

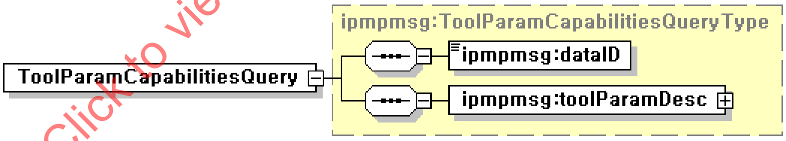
### 6.3.3.2 GetToolsResponse

This message is sent by the IPMP Processor to an IPMP Tool in response to a GetTools message.

| | |
|---|---|
| **Diagram** |  ipmpmsg:GetToolsResponseType<br>GetToolsResponse<br>ipmpmsg:dataID<br>ipmpmsg:Tool<br>0..∞ |
| **Children** | `<dataID>` `<Tool>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetToolsResponse" type="ipmpmsg:GetToolsResponseType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolsResponseType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element ref="ipmpmsg:Tool" minOccurs="0" maxOccurs="unbounded"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

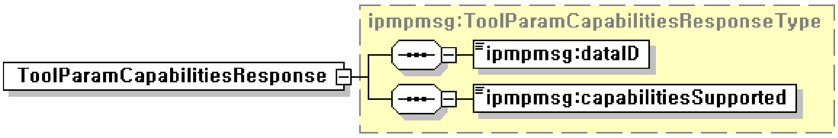
### 6.3.3.3 ToolParamCapabilitiesQuery

This message allows an IPMP Processor or an IPMP Tool to query another IPMP Tool if specific functionalities described by means of a parametric description are supported.

| | |
|---|---|
| **Diagram** |  ipmpmsg:ToolParamCapabilitiesQueryType<br>ToolParamCapabilitiesQuery<br>ipmpmsg:dataID<br>ipmpmsg:toolParamDesc |
| **Children** | `<dataID>` `<toolParamDesc>` |
| **IP** | IP |
| **Profile** | X |
| **Source** | `<element name="ToolParamCapabilitiesQuery" type="ipmpmsg:ToolParamCapabilitiesQueryType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="ToolParamCapabilitiesQueryType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="toolParamDesc" type="ipmpmsg:ParametricDescriptionType"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

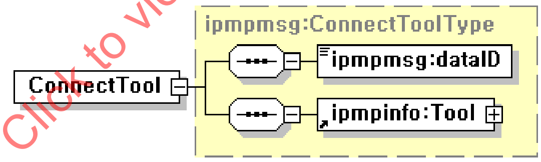
### 6.3.3.4 ToolParamCapabilitiesResponse

This message is sent in response to the above ToolParamCapabilitiesQuery message. It returns a boolean value indicating whether or not the functionalities described by means of the parametric description are supported by the Tool.

| | |
|---|---|
| **Diagram** |   ipmpmsg:ToolParamCapabilitiesResponseType<br>ToolParamCapabilitiesResponse<br>ipmpmsg:dataID<br>ipmpmsg:capabilitiesSupported |
| **Children** | `<dataID> <capabilitiesSupported >` |
| **IP** | IP |
| **Profile** | X |
| **Source** | <element name="ToolParamCapabilitiesResponse"<br>type="ipmpmsg:ToolParamCapabilitiesResponseType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="ToolParamCapabilitiesResponseType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="capabilitiesSupported" type="boolean"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

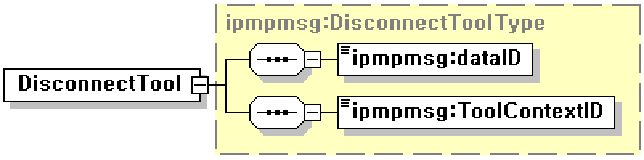
### 6.3.3.5 ConnectTool

This message, sent by an IPMP Tool to the IPMP Processor, requests the latter to create a connection to an IPMP Tool identified by the ipmpinfo:Tool element.

| | |
|---|---|
| **Diagram** |   ipmpmsg:ConnectToolType<br>ConnectTool<br>ipmpmsg:dataID<br>ipmpinfo:Tool |
| **Children** | `<dataID> < Tool>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | <element name="ConnectTool" type="ipmpmsg:ConnectToolType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="ConnectToolType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element ref="ipmpinfo:Tool"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

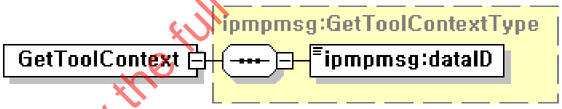
### 6.3.3.6 DisconnectTool

This message, sent by an IPMP Tool to the IPMP Processor, requests the latter to disconnect an IPMP Tool it has previously request to be connected with.

| | |
|---|---|
| **Diagram** |  ipmpmsg:DisconnectToolType<br>DisconnectTool<br>ipmpmsg:dataID<br>ipmpmsg:ToolContextID |
| **Children** | <dataID> < ToolContextID> |
| **IP** | IP |
| **Profile** | B |
| **Source** | <element name="DisconnectTool" type="ipmpmsg:DisconnectToolType" substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="DisconnectToolType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="ToolContextID" type="unsignedInt"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

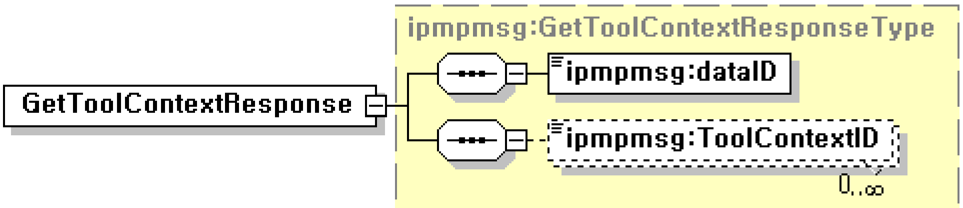
### 6.3.3.7 GetToolContext

This message is sent by an IPMP Tool to the IPMP Processor to request the list of IPMP Tool context IDs of all IPMP Tools operating on the same protected content element.

| | |
|---|---|
| **Diagram** |  ipmpmsg:GetToolContextType<br>GetToolContext<br>ipmpmsg:dataID |
| **Children** | <dataID> |
| **IP** | IP |
| **Profile** | B |
| **Source** | <element name="GetToolContext" type="ipmpmsg:GetToolContextType" substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="GetToolContextType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"/><br>    </complexContent><br></complexType> |

### 6.3.3.8 GetToolContextResponse

This message is sent by the IPMP Processor to an IPMP Tool to communicate the list of IPMP Tool context IDs of all the IPMP Tools operating on the same protected content element.

| | |
|---|---|
| **Diagram** |  ipmpmsg:GetToolContextResponseType<br>GetToolContextResponse<br>ipmpmsg:dataID<br>ipmpmsg:ToolContextID<br>0..∞ |
| **Children** | <dataID> <ToolContextID> |

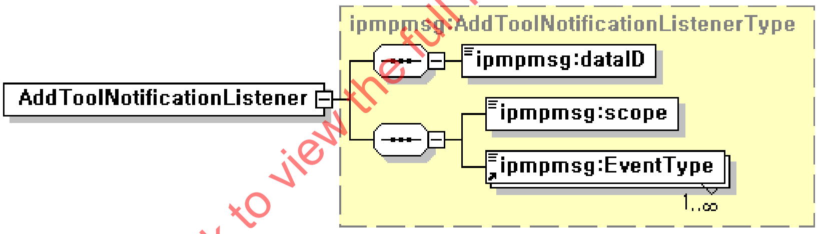
| | |
|---|---|
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetToolContextResponse"`<br>`type="ipmpmsg:GetToolContextResponseType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolContextResponseType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="ToolContextID" type="unsignedInt" minOccurs="0"`<br>`maxOccurs="unbounded"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

### 6.3.4 IPMP Tool Notification Messages

The following table specifies the IPMP Messages to be exchanged between an IPMP Processor and an IPMP Tool or a Resource Processor when being notified of particular events related to the IPMP Tools or the IPMP Processor on the device.

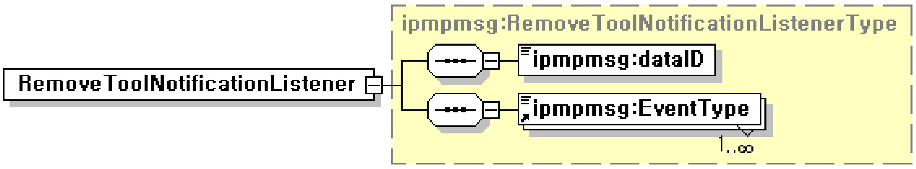
#### 6.3.4.1 AddToolNotificationListener

This message is sent by an IPMP Tool to the IPMP Processor to request of being notified when certain events occur.

| | |
|---|---|
| **Diagram** |  AddToolNotificationListener<br>ipmpmsg:AddToolNotificationListenerType<br>ipmpmsg:dataID<br>ipmpmsg:scope<br>ipmpmsg:EventType<br>1..∞ |
| **Children** | `<dataID> <scope> <EventType>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="AddToolNotificationListener"`<br>`type="ipmpmsg:AddToolNotificationListenerType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="AddToolNotificationListenerType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="scope" type="string"/>`<br>`                <element ref="ipmpmsg:EventType" maxOccurs="unbounded"/>`<br>`                <!--see scope list in MPEG IPMPX spec-->`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

### 6.3.4.2 RemoveToolNotificationListener

This message is sent by an IPMP Tool to the IPMP Processor to stop being notified when certain events occur.

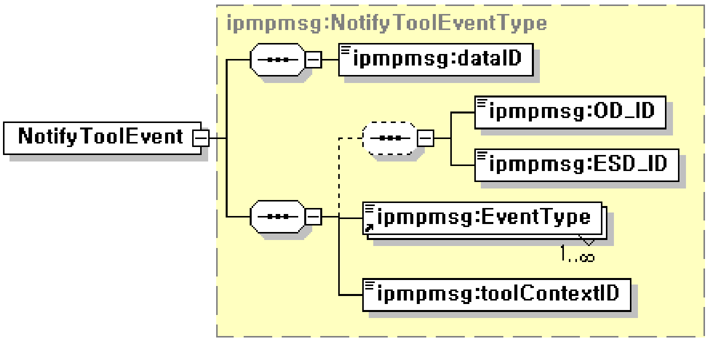
| | |
|---|---|
| **Diagram** |  ipmpmsg:RemoveToolNotificationListenerType<br>RemoveToolNotificationListener<br>ipmpmsg:dataID<br>ipmpmsg:EventType<br>1..∞ |
| **Children** | `<dataID> <scope> <EventType>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="RemoveToolNotificationListener"`<br>`type="ipmpmsg:RemoveToolNotificationListenerType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="RemoveToolNotificationListenerType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element ref="ipmpmsg:EventType" maxOccurs="unbounded"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

### 6.3.4.3 NotifyToolEvent

This message is sent by an IPMP Processor to notify an IPMP Tool that an event for which it had previously registered as a listener has occurred. The type of occurred event is conveyed by the EventType element. A list of events is given in the table below.

**Table 3 — List of Tool Events**

| Tool Event | Descriptions |
|---|---|
| 00 | CONNECTED |
| 01 | CONNECTION_FAILED |
| 02 | DISCONNECTED |
| 03 | DISCONNECTION_FAILED |
| 04 | WATERMARKDETECTED |
| 05 | PARSE_TOOLPACKDATA_SUCCESS |
| 06 | PARSE_TOOLPACKDATA_FAILED |
| 07 | UNABLE_TO_PROCESS |
| 08 | TOOL_GROUP_NOT_FOUND |
| 09 | TERMINATION_FAILED |
| 10 | CONTROLPOINT_NOT_SUPPORTED |
| 11 | UNABLE_TO_PARSE_LICENSE |
| 12 | NO_VALID_LICENSE |
| 13 | LICENSE_VALIDATION_FAILED |
| 14 | READY_TO_PLAY |
| 15 | READY_TO_BE_TERMINATED |

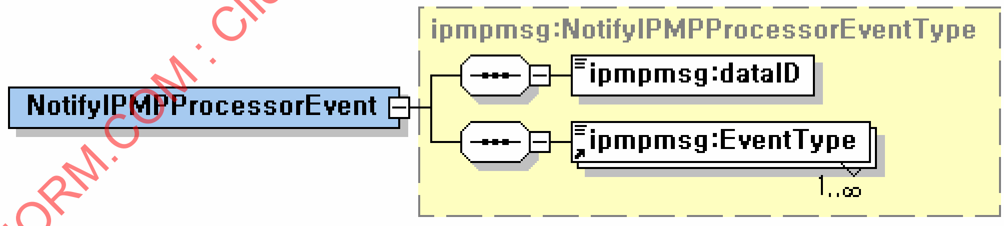
<table>
<tr><td>Diagram</td><td>

ipmpmsg:NotifyToolEventType<br>NotifyToolEvent<br>ipmpmsg:dataID<br>ipmpmsg:OD_ID<br>ipmpmsg:ESD_ID<br>ipmpmsg:EventType<br>1..∞<br>ipmpmsg:toolContextID

</td></tr>
<tr><td>Children</td><td>&lt;dataID&gt; &lt;OD_ID&gt; &lt;ESD_ID&gt; &lt;EventType&gt; &lt;toolContextID&gt;</td></tr>
<tr><td>IP</td><td>IP</td></tr>
<tr><td>Profile</td><td>B</td></tr>
<tr><td>Source</td><td><pre>
&lt;element name="NotifyToolEvent" type="ipmpmsg:NotifyToolEventType"
substitutionGroup="ipmpmsg:Data_BaseClass"/&gt;
&lt;complexType name="NotifyToolEventType"&gt;
    &lt;complexContent&gt;
        &lt;extension base="ipmpmsg:Data_BaseClassType"&gt;
            &lt;sequence&gt;
                &lt;sequence minOccurs="0"&gt;
                    &lt;element name="OD_ID" type="unsignedInt"/&gt;
                    &lt;element name="ESD_ID" type="unsignedInt"/&gt;
                &lt;/sequence&gt;
                &lt;element ref="ipmpmsg:EventType" maxOccurs="unbounded"/&gt;
                &lt;element name="toolContextID" type="unsignedInt"/&gt;
            &lt;/sequence&gt;
        &lt;/extension&gt;
    &lt;/complexContent&gt;
&lt;/complexType&gt;
</pre></td></tr>
</table>

### 6.3.4.4 NotifyIPMPProcessorEvent

This message is sent by the IPMP Processor to the Resource Processor to notify that an IPMP Processor event has occurred.

<table>
<tr><td>Diagram</td><td>

ipmpmsg:NotifyIPMPProcessorEventType<br>NotifyIPMPProcessorEvent<br>ipmpmsg:dataID<br>ipmpmsg:EventType<br>1..∞

</td></tr>
<tr><td>Children</td><td>&lt;dataID&gt;&lt;EventType&gt;</td></tr>
<tr><td>IP</td><td>IP</td></tr>
<tr><td>Profile</td><td>B</td></tr>
<tr><td>Source</td><td><pre>
&lt;element name="NotifyIPMPProcessorEvent"
type="ipmpmsg:NotifyIPMPProcessorEventType"
substitutionGroup="ipmpmsg:Data_BaseClass"/&gt;
&lt;complexType name="NotifyIPMPProcessorEventType"&gt;
    &lt;complexContent&gt;
        &lt;extension base="ipmpmsg:Data_BaseClassType"&gt;
            &lt;sequence&gt;
                &lt;element ref="ipmpmsg:EventType" maxOccurs="unbounded"/&gt;
            &lt;/sequence&gt;
        &lt;/extension&gt;
    &lt;/complexContent&gt;
&lt;/complexType&gt;
</pre></td></tr>
</table>

By employing this message, the IPMP Processor can notify whether an event has occurred, such as a connection error or a license parsing error. The NotifyIPMPProcessorEvent message extends the Data_BaseClass Type by conveying:

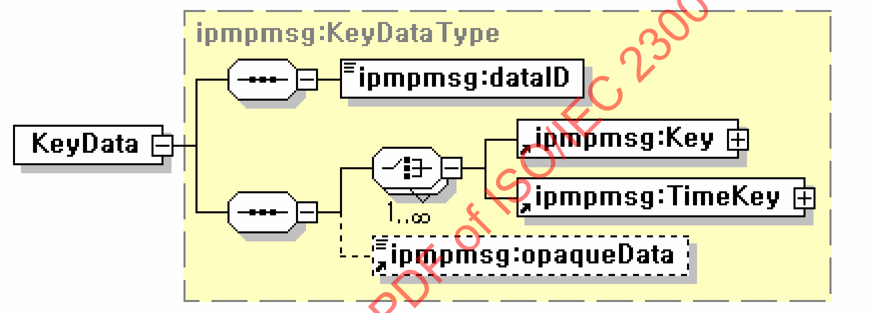
- EventType: This element specifies the type of the event to notify. A list of events is given in Table 2.

### 6.3.5 IPMP Processing Messages

The following table specifies the IPMP Messages to be exchanged between two IPMP Tools or between an IPMP Processor and an IPMP Tool or between an IPMP Processor and a Resource Processor for different purposes.

#### 6.3.5.1 KeyData

This message conveys either a time-dependent or a time-independent decryption key.

| | |
|---|---|
| **Diagram** |  ipmpmsg:KeyDataType<br>KeyData<br>ipmpmsg:dataID<br>ipmpmsg:Key<br>ipmpmsg:TimeKey<br>1..∞<br>ipmpmsg:opaqueData |
| **Children** | <dataID> <Key> <TimeKey> <OpaqueData> |
| **IP** | – |
| **Profile** | B |
| **Source** | <element name="KeyData" type="ipmpmsg:KeyDataType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="KeyDataType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <choice maxOccurs="unbounded"><br>                    <element ref="ipmpmsg:Key"/><br>                    <element ref="ipmpmsg:TimeKey"/><br>                </choice><br>                <element name="opaqueData" type="base64Binary" minOccurs="0"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType><br><complexType name="KeyBaseType" abstract="true"/><br><element name="Key" type="ipmpmsg:KeyType"/><br><complexType name="KeyType"><br>    <complexContent><br>        <extension base="ipmpmsg:KeyBaseType"><br>            <sequence><br>                <choice><br>                    <element ref="xenc:EncryptedData"/><br>                    <element ref="xenc:EncryptedKey"/><br>                </choice><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType><br><element name="TimeKey" type="ipmpmsg:TimeKeyType"/><br><complexType name="TimeKeyType"><br>    <complexContent><br>        <extension base="ipmpmsg:KeyType"> |

| | |
|---|---|
| | &lt;sequence&gt;<br>&lt;element name="startDTS" type="unsignedLong" minOccurs="0"/&gt;<br>&lt;element name="startPacketID" type="unsignedInt" minOccurs="0"/&gt;<br>&lt;element name="expireDTS" type="unsignedLong" minOccurs="0"/&gt;<br>&lt;element name="expirePacketID" type="unsignedInt" minOccurs="0"/&gt;<br>&lt;/sequence&gt;<br>&lt;/extension&gt;<br>&lt;/complexContent&gt;<br>&lt;/complexType&gt; |

If the key has no time constraints, it will be conveyed within the ipmpmsg:Key element, otherwise the ipmpmsg:TimeKey element is employed.

The ipmpinfo:Key element conveys a decryption key and further data in its child elements xenc:EncryptedData and xenc:EncryptedKey.

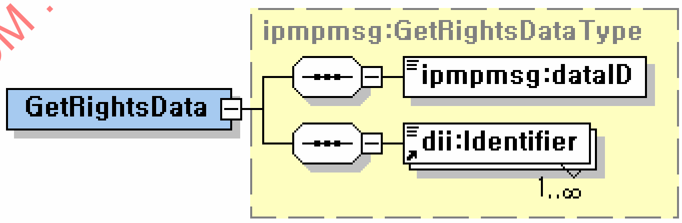
The ipmpinfo:TimeKey element extends the ipmpmsg:KeyType element by conveying further information about the use of the key by the IPMP Tool that received it.

- startDTS: the container of the decoding time stamp of the first access unit for which the contained key is valid (optional)
- startPacketID: the container of the packet ID of the first access unit for which the contained key is valid (optional)
- expireDTS: the container for the decoding time stamp of the first access unit for which the contained key is no longer valid (optional)
- expirePacketID: the container for the packet ID of the first access unit for which the contained key is no longer valid (optional)

Whenever a Key has to be delivered to an IPMP Tool, this is done by employing a KeyData message.

### 6.3.5.2 GetRightsData

This message is sent by an IPMP Tool or by the IPMP Processor to the Resource Processor to request rights information (i.e. a license) related to a specific content item or any parts of it.

| | |
|---|---|
| **Diagram** |  ipmpmsg:GetRightsDataType<br>GetRightsData<br>ipmpmsg:dataID<br>dii:Identifier<br>1..∞ |
| **Children** | &lt;dataID&gt;&lt;Identifier&gt; |
| **IP** | **IP** |
| **Profile** | **B** |
| **Source** | &lt;element name="GetRightsData" type="ipmpmsg:GetRightsDataType" substitutionGroup="ipmpmsg:Data_BaseClass"/&gt;<br>&lt;complexType name="GetRightsDataType"&gt;<br>&lt;complexContent&gt;<br>&lt;extension base="ipmpmsg:Data_BaseClassType"&gt;<br>&lt;sequence&gt;<br>&lt;element ref="dii:Identifier" maxOccurs="unbounded"/&gt;<br>&lt;/sequence&gt;<br>&lt;/extension&gt;<br>&lt;/complexContent&gt;<br>&lt;/complexType&gt; |

The entity receiving a GetRightsData message shall reply with a RightsData message.

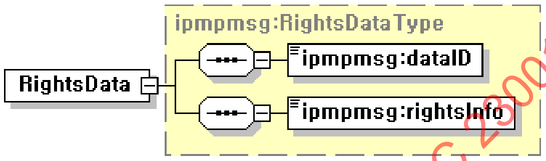
When the IPMP Processor requires rights information related to a content item or parts thereof in order to operate, the IPMP Processor requests it by employing the GetRightsData message. The IPMP Processor specifies the identifier of the content item/content element for which the license is requested.

The GetRightsData message extends the Data_BaseClass Type by conveying the following element:

- Identifier: This element specifies the content or content element identifier for which rights information is requested

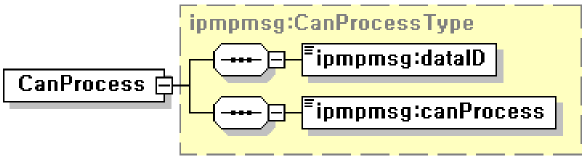
### 6.3.5.3 RightsData

The RightsData message is employed to deliver a license [2] to an IPMP Tool.

| | |
|---|---|
| **Diagram** |   ipmpmsg:RightsDataType<br>RightsData<br>ipmpmsg:dataID<br>ipmpmsg:rightsInfo |
| **Children** | <dataID> <RightsInfo> |
| **IP** | – |
| **Profile** | B |
| **Source** | `<element name="RightsData" type="ipmpmsg:RightsDataType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="RightsDataType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="rightsInfo" type=" r:License"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

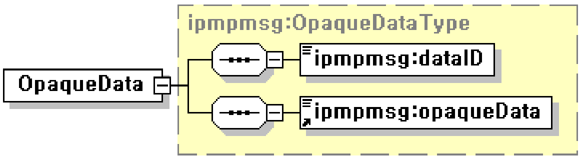
### 6.3.5.4 CanProcess

This message is sent by an IPMP Tool to the IPMP Processor to allow or deny content processing.

| | |
|---|---|
| **Diagram** |   ipmpmsg:CanProcessType<br>CanProcess<br>ipmpmsg:dataID<br>ipmpmsg:canProcess |
| **Children** | <dataID> <canProcess> |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="CanProcess" type="ipmpmsg:CanProcessType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="CanProcessType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="canProcess" type="boolean"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

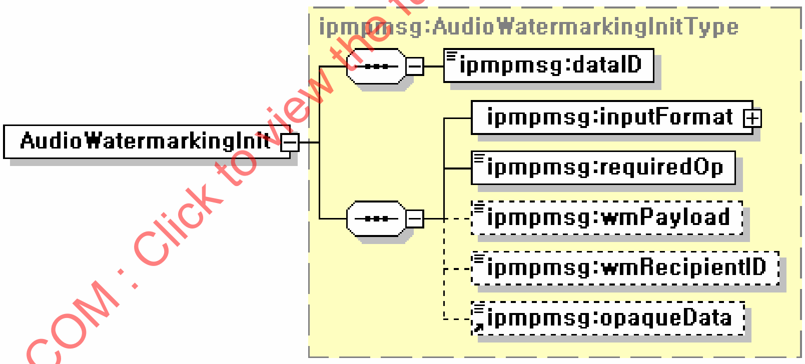
#### 6.3.5.5 OpaqueData

This is a general-purpose container message for any type of data.

| | |
|---|---|
| **Diagram** |   ipmpmsg:OpaqueDataType<br>OpaqueData<br>ipmpmsg:dataID<br>ipmpmsg:opaqueData |
| **Children** | <dataID> <opaqueData> |
| **IP** | – |
| **Profile** | B |
| **Source** | <element name="OpaqueData" type="ipmpmsg:OpaqueDataType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="OpaqueDataType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="opaqueData" type="base64Binary"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

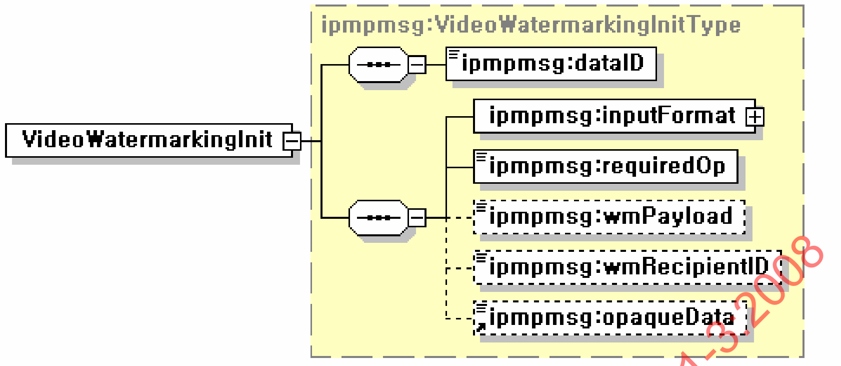
#### 6.3.5.6 AudioWatermarkingInit

This message is addressed to an IPMP Tool performing audio watermarking, delivering specific information about the audio content and the operation to perform on it.

| | |
|---|---|
| **Diagram** |   ipmpmsg:AudioWatermarkingInitType<br>AudioWatermarkingInit<br>ipmpmsg:dataID<br>ipmpmsg:inputFormat<br>ipmpmsg:requiredOp<br>ipmpmsg:wmPayload<br>ipmpmsg:wmRecipientID<br>ipmpmsg:opaqueData |
| **Children** | <dataID> <inputFormat> <requiredOp> <wmPayload> <wmRewcipientID> <opaqueData> |
| **IP** | – |
| **Profile** | X |
| **Source** | <element name="AudioWatermarkingInit" type="ipmpmsg:AudioWatermarkingInitType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="AudioWatermarkingInitType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="inputFormat" type="ipmpmsg:AudioFormatType"/><br>                <element name="requiredOp" type="ipmpmsg:requiredOpType"/><br>                <element name="wmPayload" type="string" minOccurs="0"/><br>                <element name="wmRecipientID" type="anyURI" minOccurs="0"/><br>                <element name=" opaqueData" type="base64Binary"<br>minOccurs="0"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

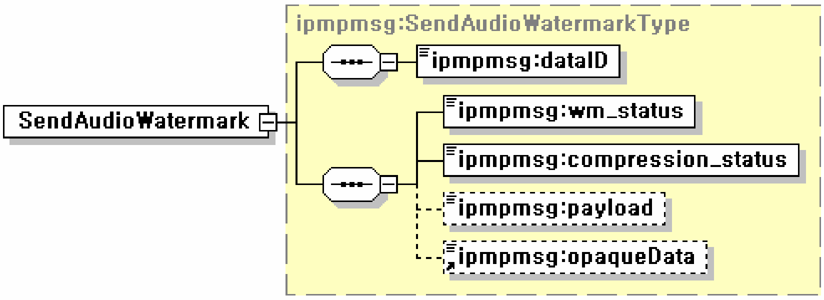
### 6.3.5.7 VideoWatermarkingInit

This message is addressed to an IPMP Tool performing video watermarking, delivering specific information about the video content and the operation that shall be perform on it.

| | |
|---|---|
| **Diagram** |  ipmpmsg:VideoWatermarkingInitType<br>VideoWatermarkingInit<br>ipmpmsg:dataID<br>ipmpmsg:inputFormat<br>ipmpmsg:requiredOp<br>ipmpmsg:wmPayload<br>ipmpmsg:wmRecipientID<br>ipmpmsg:opaqueData |
| **Children** | <dataID> <inputFormat> <requiredOp> <wmPayload> <wmRewcipientID> <opaqueData> |
| **IP** | – |
| **Profile** | X |
| **Source** | <element name="VideoWatermarkingInit"<br>type="ipmpmsg:VideoWatermarkingInitType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="VideoWatermarkingInitType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="inputFormat" type="ipmpmsg:VideoFormatType"/><br>                <element name="requiredOp" type="ipmpmsg:requiredOpType"/><br>                <element name="wmPayload" type="string" minOccurs="0"/><br>                <element name="wmRecipientID" type="anyURI" minOccurs="0"/><br>                <element name=" opaqueData" type="base64Binary"<br>minOccurs="0"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

### 6.3.5.8 SendAudioWatermark

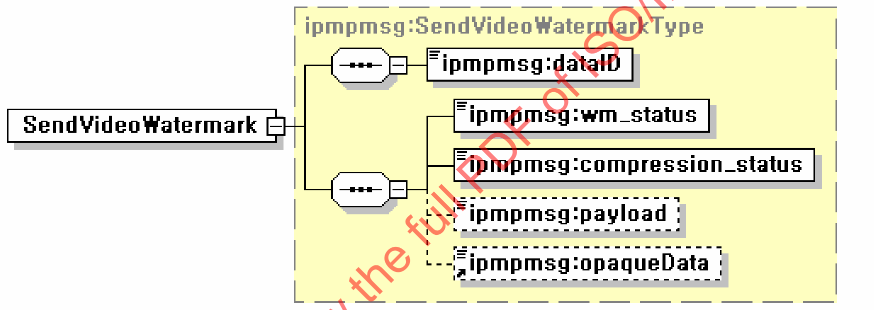
This message is sent from an audio watermarking IPMP Tool to another IPMP Tool conveying a set of information as a result of an audio watermarking operation.

| | |
|---|---|
| **Diagram** |  ipmpmsg:SendAudioWatermarkType<br>SendAudioWatermark<br>ipmpmsg:dataID<br>ipmpmsg:wm_status<br>ipmpmsg:compression_status<br>ipmpmsg:payload<br>ipmpmsg:opaqueData |
| **Children** | <dataID> <wm_status> <compression_status> <payload> <opaqueData> |
| **IP** | – |

| | |
|---|---|
| **Profile** | x |
| **Source** | <element name="SendAudioWatermark" type="ipmpmsg:SendAudioWatermarkType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="SendAudioWatermarkType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="wm_status" type="ipmpmsg:wm_statusType"/><br>                <element name="compression_status"<br>type="ipmpmsg:compression_statusType"/><br>                <element name="payload" type="string" minOccurs="0"/><br>                <element name=" opaqueData" type="base64Binary"<br>minOccurs="0"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

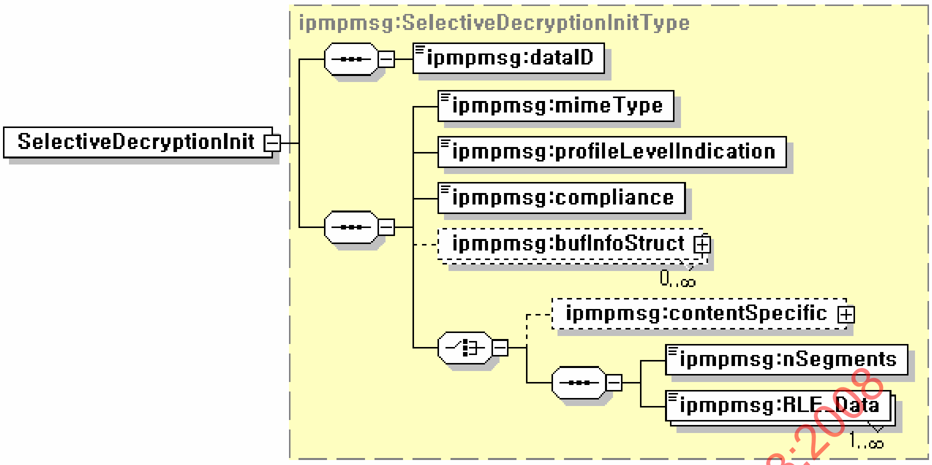
### 6.3.5.9 SendVideoWatermark

This message is sent by a video watermarking IPMP Tool to another IPMP Tool conveying a set of information as a result of a video watermarking operation.

| | |
|---|---|
| **Diagram** |  ipmpmsg:SendVideoWatermarkType<br>SendVideoWatermark<br>ipmpmsg:dataID<br>ipmpmsg:wm_status<br>ipmpmsg:compression_status<br>ipmpmsg:payload<br>ipmpmsg:opaqueData |
| **Children** | <dataID> <wm_status> <compression_status> <payload> <opaqueData> |
| **IP** | – |
| **Profile** | x |
| **Source** | <element name="SendVideoWatermark" type="ipmpmsg:SendVideoWatermarkType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="SendVideoWatermarkType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="wm_status" type="ipmpmsg:wm_statusType"/><br>                <element name="compression_status"<br>type="ipmpmsg:compression_statusType"/><br>                <element name="payload" type="string" minOccurs="0"/><br>                <element name="opaqueData" type="base64Binary" minOccurs="0"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

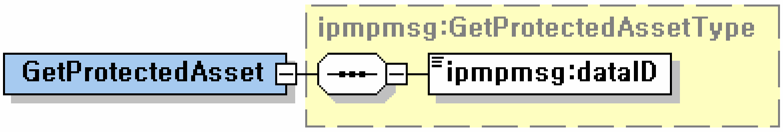
### 6.3.5.10 SelectiveDecryptionInit

This message is used to communicate to an IPMP Tool performing decryption how the encryption on the received content bitstream has been applied, whether the whole bitstream has been encrypted or only portions of it. In the case in which only portions of the received bitstreams are encrypted, the information allowing to identify the portions to be decrypted is conveyed by this message.

| | |
|---|---|
| **Diagram** |  ipmpmsg:SelectiveDecryptionInitType<br>SelectiveDecryptionInit<br>ipmpmsg:dataID<br>ipmpmsg:mimeType<br>ipmpmsg:profileLevelIndication<br>ipmpmsg:compliance<br>ipmpmsg:bufInfoStruct<br>0..∞<br>ipmpmsg:contentSpecific<br>ipmpmsg:nSegments<br>ipmpmsg:RLE_Data<br>1..∞ |
| **Children** | <dataID> <mimeType> <profileLevelIndication> <compliance> <bufInfostruct> <contentSpecific> <nSegments> <RLE_Data> |
| **IP** | – |
| **Profile** | X |
| **Source** | <element name="SelectiveDecryptionInit"<br>type="ipmpmsg:SelectiveDecryptionInitType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="SelectiveDecryptionInitType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <element name="mimeType" type="string"/><br>                <element name="profileLevelIndication" type="string"/><br>                <element name="compliance" type="string"/><br>                <element name="bufInfoStruct" type="ipmpmsg:bufInfoStructType"<br>minOccurs="0" maxOccurs="unbounded"/><br>                <choice><br>                    <element name="contentSpecific"<br>type="ipmpmsg:contentSpecificType" minOccurs="0"/><br>                    <sequence><br>                        <element name="nSegments" type="short"/><br>                        <element name="RLE_Data" type="short"<br>maxOccurs="unbounded"/><br>                    </sequence><br>                </choice><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

### 6.3.5.11 GetProtectedAsset

This message is sent by the IPMP Processor to the Resource Processor requesting a protected asset.

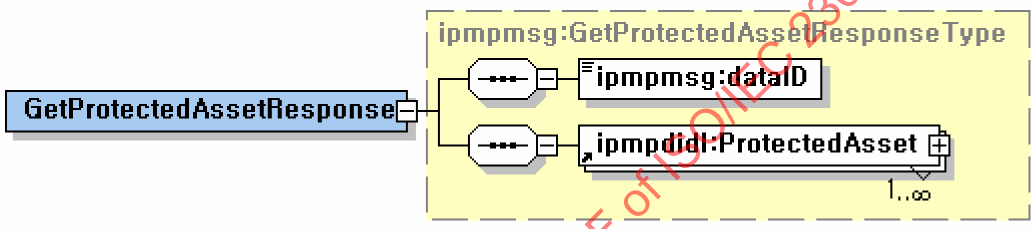
| | |
|---|---|
| **Diagram** |  ipmpmsg:GetProtectedAssetType<br>GetProtectedAsset<br>ipmpmsg:dataID |
| **Children** | <dataID> |
| **IP** | IP |

| | |
|---|---|
| **Profile** | B |
| **Source** | `<element name="GetProtectedAsset" type="ipmpmsg:GetProtectedAssetType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetProtectedAssetType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType"/>`<br>`    </complexContent>`<br>`</complexType>` |

Upon receiving this message, the Resource Processor sends a GetProtectedAssetResponse message in response.

### 6.3.5.12 GetProtectedAssetResponse

This message is sent by a Resource Processor to an IPMP Processor to transmit a ProtectedAsset.

| | |
|---|---|
| **Diagram** |  GetProtectedAssetResponse<br>ipmpmsg:GetProtectedAssetResponseType<br>ipmpmsg:dataID<br>ipmpdidl:ProtectedAsset<br>1..∞ |
| **Children** | `<dataID><ProtectedAsset>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetProtectedAssetResponse" type="ipmpmsg:GetProtectedAssetResponseType"/>`<br>`<complexType name="GetProtectedAssetResponseType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element ref="ipmpdidl:ProtectedAsset" maxOccurs="unbounded"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

Once a Resource Processor receives a GetProtectedAsset message, the Resource Processor sends a GetProtectedAssetResponse message in response. The GetProtectedAssetResponse message extends the Data_BaseClassType by conveying the following element:

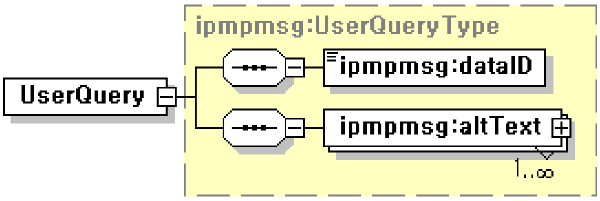
- ProtectedAsset: this element contains IPMP information related to a protected content item/content element. The IPMP Processor can apply tools to the contents based on the IPMP information contained within this element.

### 6.3.6 User Interaction Messages

This section specifies the IPMP Messages enabling the communication between an IPMP Tool and the user of the device on which the IPMP Tool operates.

### 6.3.6.1 UserQuery

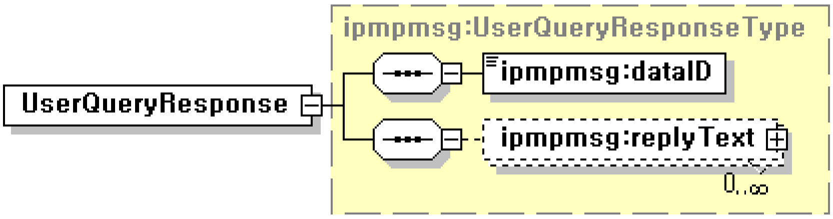
Message used to query the User for information.

| | |
|---|---|
| **Diagram** |  ipmpmsg:UserQueryType<br>UserQuery<br>ipmpmsg:dataID<br>ipmpmsg:altText<br>1..∞ |
| **Children** | <dataID> <altText> |
| **IP** | IP |
| **Profile** | B |
| **Source** | |

```
<element name="UserQuery" type="ipmpmsg:UserQueryType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="UserQueryType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType">
                <sequence>
                    <element name="altText" type="ipmpmsg:altTextType"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="altTextType">
        <sequence>
            <element name="languageCode" type="language"/>
            <element name="titleText" type="string" minOccurs="0"/>
            <element name="displayText" type="ipmpmsg:displayTextType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="needReplyText" type="ipmpmsg:replyTextType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="inclOptionSelect"
type="ipmpmsg:inclOptionSelectType" minOccurs="0" maxOccurs="unbounded"/>
            <element name="SMIL_URL" type="string" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="SMIL" type="hexBinary" minOccurs="0"/>
            <!-- ISO 639-2:1998 bibliographic three character language code-->
        </sequence>
    </complexType>
    <complexType name="displayTextType">
        <sequence>
            <element name="text" type="string"/>
        </sequence>
        <attribute name="ID" type="unsignedShort" use="required"/>
    </complexType>
    <complexType name="replyTextType">
        <sequence>
            <element name="text" type="string"/>
        </sequence>
        <attribute name="ID" type="unsignedShort" use="required"/>
        <attribute name="subID" type="unsignedShort" use="required"/>
        <attribute name="isHidden" type="boolean"/>
    </complexType>
    <complexType name="inclOptionSelectType">
        <sequence>
            <element name="promptText" type="string"/>
        </sequence>
        <attribute name="ID" type="unsignedShort" use="required"/>
        <attribute name="subID" type="unsignedShort" use="required"/>
        <attribute name="isExclusive" type="boolean"/>
    </complexType>
```

#### 6.3.6.2 UserQueryResponse

This message contains the user response to a UserQuery message.

| | |
|---|---|
| **Diagram** |   ipmpmsg:UserQueryResponseType<br>UserQueryResponse<br>ipmpmsg:dataID<br>ipmpmsg:replyText<br>0..∞ |
| **Children** | `<dataID> <replyText>` |
| **IP** | **IP** |
| **Profile** | **B** |
| **Source** | `<element name="UserQueryResponse" type="ipmpmsg:UserQueryResponseType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`    <complexType name="UserQueryResponseType">`<br>`        <complexContent>`<br>`            <extension base="ipmpmsg:Data_BaseClassType">`<br>`                <sequence>`<br>`                    <element name="replyText"`<br>`type="ipmpmsg:responseReplyTextType" minOccurs="0" maxOccurs="unbounded"/>`<br>`                </sequence>`<br>`            </extension>`<br>`        </complexContent>`<br>`    </complexType>`<br>`    <complexType name="responseReplyTextType">`<br>`        <sequence>`<br>`            <element name="languageCode" type="language"/>`<br>`            <element name="replyText" type="ipmpmsg:replyTextType"`<br>`minOccurs="0" maxOccurs="unbounded"/>`<br>`            <element name="optionResult" type="ipmpmsg:optionResultType"`<br>`minOccurs="0" maxOccurs="unbounded"/>`<br>`            <!-- ISO 639-2:1998 bibliographic three character language code-->`<br>`        </sequence>`<br>`    </complexType>`<br>`    <complexType name="optionResultType">`<br>`        <sequence>`<br>`            <element name="result" type="boolean"/>`<br>`        </sequence>`<br>`        <attribute name="ID" type="unsignedShort" use="optional"/>`<br>`        <attribute name="subID" type="unsignedShort" use="optional"/>`<br>`    </complexType>` |

### 6.3.7 Additional IPMP Messages

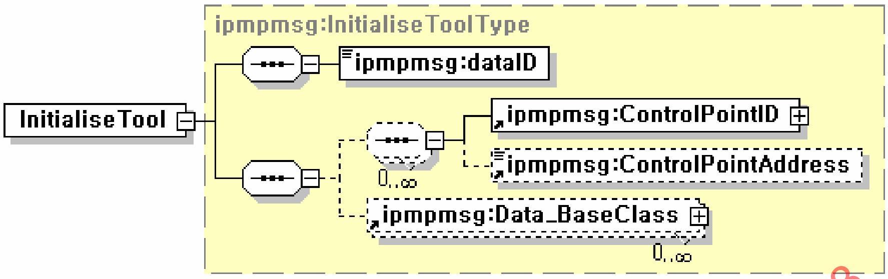
This section specifies additional IPMP Messages to be exchanged between an IPMP Tool and an IPMP Processor or between an IPMP Processor and a Resource Processor for different purposes.

Note: a compliant single IPMP Tool does not need to be able to process IPMP Tool Pack messages.

### 6.3.7.1 InitialiseTool

This message conveys initialisation data to an IPMP Tool.

<table>
<tr><td><b>Diagram</b></td><td>



</td></tr>
<tr><td><b>Children</b></td><td><dataID> <ControlPointID> <ControlPointAddress> <Data_BaseClass></td></tr>
<tr><td><b>IP</b></td><td>IP</td></tr>
<tr><td><b>Profile</b></td><td>B</td></tr>
<tr><td><b>Source</b></td><td>

```
<element name="InitialiseTool" type="ipmpmsg:InitialiseToolType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="InitialiseToolType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType">
                <sequence>
                    <sequence minOccurs="0" maxOccurs="unbounded">
                        <element ref="ipmpmsg:ControlPointID"/>
                        <element ref="ipmpmsg:ControlPointAddress"
minOccurs="0"/>
                    </sequence>
                    <element name="Data_BaseClass" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="ControlPointID" type="ipmpmsg:ControlPointType"/>
    <complexType name="ControlPointType">
        <sequence>
            <element name="ID" type="integer"/>
        </sequence>
    </complexType>
    <element name="ControlPointAddress" type="base64Binary"/>
```

</td></tr>
</table>

An ipmpmsg:InitialiseTool element delivers the following information:

- ControlPointID: an ID indicating the point where the IPMP Tool can operate along the Resource processing pipeline. This ID is registered by the appointed registration agency before being used. The preliminary ipmpmsg:ControlPointID assignments are given in Table 4:

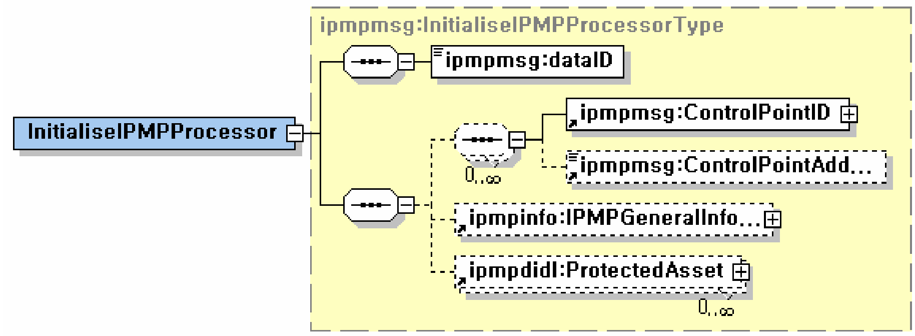
**Table 4 — List of available Control Points**

| ControlPointID | Description |
|---|---|
| 00 | NO_CONTROL_POINT |
| 01 | CONTROL_POINT_BEFORE_DEMULTIPLEXING |
| 02 | CONTROL_POINT_BEFORE_AUDIO_DECODING |
| 03 | CONTROL_POINT_AFTER_AUDIO_DECODING |
| 04 | CONTROL_POINT_BEFORE_VIDEO_DECODING |
| 05 | CONTROL_POINT_AFTER_VIDEO_DECODING |
| 06 | CONTROL_POINT_BEFORE_STORING |
| 07 | CONTROL_POINT_BEFORE_PLAYBACK |
| 08 | CONTROL_POINT_BEFORE_TRANSFERRING |

- ControlPointAddress: a local address on the device, in which the IPMP Tool can be inserted, and receive media data for processing. The format of this address may depend on the particular device on which the IPMP Processor is running, and such information depends on the Instantiation_API_ID value in the ipmpinfo-msx:ToolAPI_Config element [1].
- Any message extending ipmpmsg:Data_BaseClass.

### 6.3.7.2 InitialiseIPMPProcessor

This message is sent by a Resource Processor to initialise the IPMP Processor.

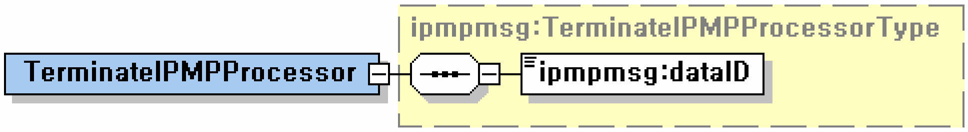
| | |
|---|---|
| **Diagram** |  ipmpmsg:InitialiseIPMPProcessorType<br>InitialiseIPMPProcessor<br>ipmpmsg:dataID<br>ipmpmsg:ControlPointID<br>ipmpmsg:ControlPointAdd...<br>0..∞<br>ipmpinfo:IPMPGeneralInfo...<br>ipmpdidl:ProtectedAsset<br>0..∞ |
| **Children** | <dataID> <ControlPointID> <ControlPointAddress> <IPMPGeneralInfoDescriptor> <ProtectedAsset> |
| **IP** | IP |
| **Profile** | B |
| **Source** | <element name="InitialiseIPMPProcessor"<br>type="ipmpmsg:InitialiseIPMPProcessorType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="InitialiseIPMPProcessorType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"><br>            <sequence><br>                <sequence minOccurs="0" maxOccurs="unbounded"><br>                    <element ref="ipmpmsg:ControlPointID"/><br>                    <element ref="ipmpmsg:ControlPointAddress" minOccurs="0"/><br>                </sequence><br>                <element ref="ipmpinfo:IPMPGeneralInfoDescriptor"<br>minOccurs="0"/><br>                <element ref="ipmpdidl:ProtectedAsset" minOccurs="0"<br>maxOccurs="unbounded"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

The InitialiseIPMPProcessor message is employed to initialise the IPMP Processor. The InitialiseIPMPProcessor message extends the Data_BaseClass Type by conveying the following information:

- ControlPointID: the list of control point identifiers on which the IPMP Tool shall operate
- ControlPointAddress: the physical address of the control points.
- IPMPGeneralInfoDescriptor: the general IPMP information descriptor such as the IPMP Tool List and possibly licenses.
- ProtectedAsset: the IPMP information required to instantiate and initialize IPMP Tools

### 6.3.7.3 TerminateIPMPProcessor

This message is sent by a Resource Processor to terminate the IPMP Processor.

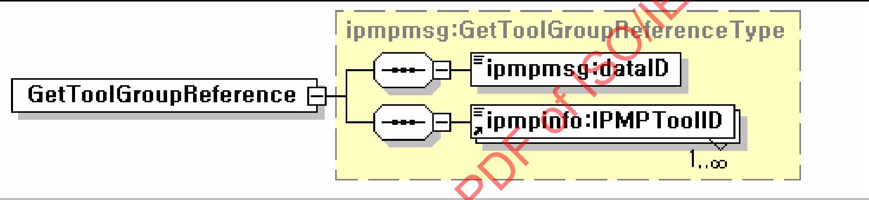
| | |
|---|---|
| **Diagram** |  ipmpmsg:TerminateIPMPProcessorType<br>TerminateIPMPProcessor<br>ipmpmsg:dataID |
| **Children** | <dataID> |
| **IP** | IP |

| | |
|---|---|
| **Profile** | B |
| **Source** | <element name="TerminateIPMPProcessor"<br>type="ipmpmsg:TerminateIPMPProcessorType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="TerminateIPMPProcessorType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"/><br>    </complexContent><br></complexType> |

The TerminateIPMPProcessor message is used to terminate the IPMP Processor. A Resource Process sends a TerminateIPMPProcessor message to the IPMP Processor when consumption of the content is over. Once the IPMP Processor receives this message, the IPMP Processor disconnects all the IPMP Tools used on the resources and terminates all its processes.

### 6.3.7.4 GetToolGroupReference

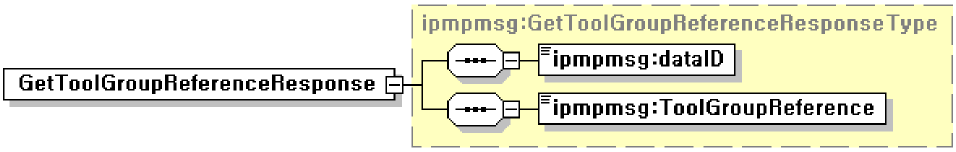
This message is sent by an IPMP Tool Agent to the IPMP Processor requesting the reference to the associated IPMP Tool Group.

| | |
|---|---|
| **Diagram** |  ipmpmsg:GetToolGroupReferenceType<br>GetToolGroupReference<br>ipmpmsg:dataID<br>ipmpinfo:IPMPToolID<br>1..∞ |
| **Children** | <dataID> |
| **IP** | IP |
| **Profile** | B |
| **Source** | <element name="GetToolGroupReference"<br>type="ipmpmsg:GetToolGroupReferenceType"<br>substitutionGroup="ipmpmsg:Data_BaseClass"/><br><complexType name="GetToolGroupReferenceType"><br>    <complexContent><br>        <extension base="ipmpmsg:Data_BaseClassType"/><br>            <sequence><br>                <element ref="ipmpinfo:IPMPToolID" maxOccurs="unbounded"/><br>            </sequence><br>    </complexContent><br></complexType> |

Each IPMP Tool Pack optionally contains the IPMP Tool Group. The IPMP Tool Agent requires the IPMP Tool Group identified by ipmpinfo:IPMPToolID to perform IPMP functions. This message is sent by the IPMP Tool Agent to request the reference of the IPMP Tool Group

### 6.3.7.5 GetToolGroupReferenceResponse

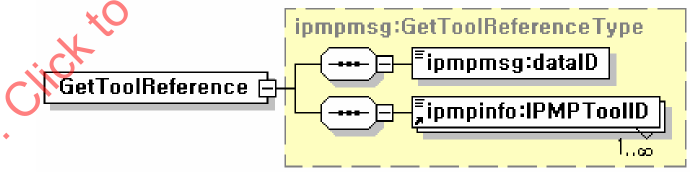
This message is sent by an IPMP Processor to an IPMP Tool Agent conveying the reference to the associated IPMP Tool Group.

| | |
|---|---|
| **Diagram** |   ipmpmsg:GetToolGroupReferenceResponseType<br>GetToolGroupReferenceResponse<br>ipmpmsg:dataID<br>ipmpmsg:ToolGroupReference |
| **Children** | <dataID> <ToolGroupReference> |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetToolGroupReferenceResponse"`<br>`type="ipmpmsg:GetToolGroupReferenceResponseType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolGroupReferenceResponseType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="ToolGroupReference" type="base64Binary"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

- ToolGroupReference: the reference to the IPMP Tool Group. The reference indicates either the handle or the local address of the IPMP Tool Group depending on the Instantiation_API_ID value in the ipmpinfo-msx:ToolAPI_Config element [1].

### 6.3.7.6 GetToolReference

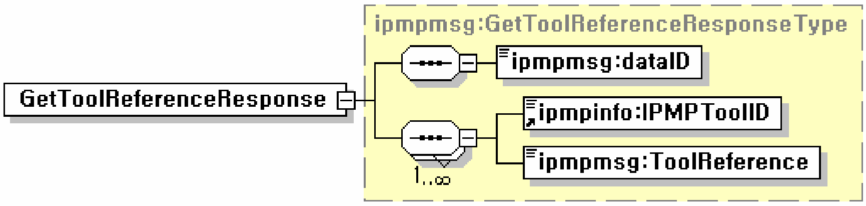
In the case the IPMP Tool Agent requires a Single IPMP Tool to perform IPMP functions, a reference to it may be obtained by sending a GetToolReference message to the IPMP Processor. This message is sent by the IPMP Tool Agent to request the reference of the Single IPMP Tool.

| | |
|---|---|
| **Diagram** |   ipmpmsg:GetToolReferenceType<br>GetToolReference<br>ipmpmsg:dataID<br>ipmpinfo:IPMPToolID<br>1..∞ |
| **Children** | <dataID> <IPMPToolID> |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetToolReference" type="ipmpmsg:GetToolReferenceType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolReferenceType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element ref="ipmpinfo:IPMPToolID" maxOccurs="unbounded"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

- IPMPToolID: The identifier of the Single IPMP Tools that the IPMP Tool Agent is requesting. More than one IPMP Tool Reference can be requested.

### 6.3.7.7 GetToolReferenceResponse

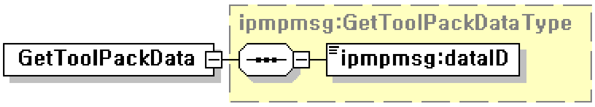
This message conveys the reference of a requested Single IPMP Tool. This message is sent by the IPMP Processor to the IPMP Tool Agent.

| | |
|---|---|
| **Diagram** |   ipmpmsg:GetToolReferenceResponseType<br>GetToolReferenceResponse<br>ipmpmsg:dataID<br>ipmpinfo:IPMPToolID<br>ipmpmsg:ToolReference<br>1..∞ |
| **Children** | `<dataID> <IPMPToolID> <ToolReference>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetToolReferenceResponse"`<br>`type="ipmpmsg:GetToolReferenceResponseType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolReferenceResponseType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence maxOccurs="unbounded">`<br>`                <element ref="ipmpinfo:IPMPToolID"/>`<br>`                <element name="ToolReference" type="base64Binary"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

- IPMPToolID: the ID of the Single IPMP Tool that the IPMP Tool Agent has requested.
- ToolReference: the reference of the Single IPMP Tool identified by the IPMPToolID. The reference indicates either the handle or the local address of the IPMP Tool Group depending on the Instantiation_API_ID value in the ToolAPI_Config [1].

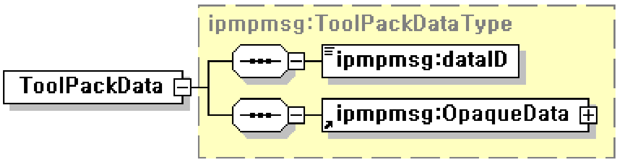
### 6.3.7.8 GetToolPackData

The IPMP Tool Agent may need Tool Pack Data to initialize the IPMP Tool Group or the Single IPMP Tool. This message is sent by the IPMP Tool Agent to request the Tool Pack Data.

| | |
|---|---|
| **Diagram** |   ipmpmsg:GetToolPackDataType<br>GetToolPackData<br>ipmpmsg:dataID |
| **Children** | `<dataID>` |
| **IP** | IP |
| **Profile** | B |
| **Source** | `<element name="GetToolPackData" type="ipmpmsg:GetToolPackDataType"`<br>`substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="GetToolPackDataType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType"/>`<br>`    </complexContent>`<br>`</complexType>` |

### 6.3.7.9 ToolPackData

This message is sent by an IPMP Processor to an IPMP Tool Agent conveying the IPMP Tool Pack Data.

| | |
|---|---|
| **Diagram** |  ipmpmsg:ToolPackDataType<br>ToolPackData<br>ipmpmsg:dataID<br>ipmpmsg:OpaqueData |
| **Children** | `<dataID> <OpaqueData>` |
| **IP** | **IP** |
| **Profile** | **B** |
| **Source** | `<element name="ToolPackData" type="ipmpmsg:ToolPackDataType" substitutionGroup="ipmpmsg:Data_BaseClass"/>`<br>`<complexType name="ToolPackDataType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:Data_BaseClassType">`<br>`            <sequence>`<br>`                <element name="opaqueData" type="base64Binary" minOccurs="0"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

- OpaqueData: the information to initialize the IPMP Tool Group or the Single IPMP Tool.

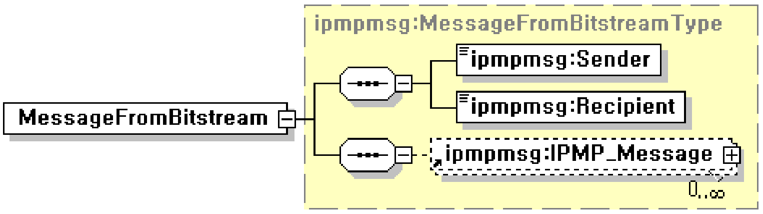
## 6.4 Legacy Messages

### 6.4.1 Introduction

Two messages defined in ISO/IEC 13818-11 and ISO/IEC 14496-13, namely IPMP_MessageFromBitstream and IPMP_DescriptorFromBitstream have been translated into XML for the sake of completeness of this specification. For the semantics of these messages, refer to ISO/IEC 13818-11 and ISO/IEC 14496-13.

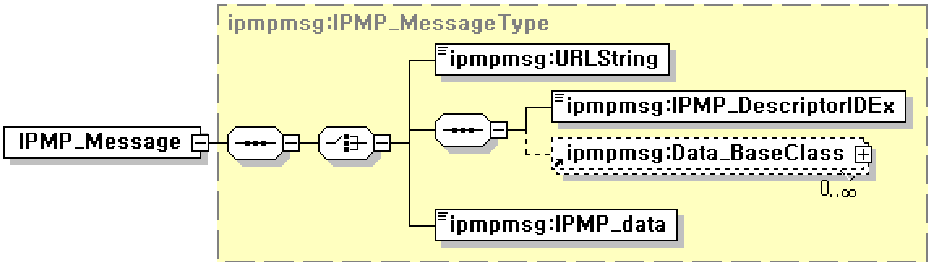
### 6.4.2 MessageFromBitstream

The purpose of a MessageFromBitstream is the delivery of IPMP_Messages to IPMP Tools.

| | |
|---|---|
| **Diagram** |  ipmpmsg:MessageFromBitstreamType<br>MessageFromBitstream<br>ipmpmsg:Sender<br>ipmpmsg:Recipient<br>ipmpmsg:IPMP_Message<br>0..∞ |
| **Children** | `<Sender> <Recipient> <IPMP_Message>` |
| **Source** | `<element name="MessageFromBitstream" type="ipmpmsg:MessageFromBitstreamType" substitutionGroup="ipmpmsg:ToolMessageBase"/>`<br>`<complexType name="MessageFromBitstreamType">`<br>`    <complexContent>`<br>`        <extension base="ipmpmsg:ToolMessageBaseType">`<br>`            <sequence>`<br>`                <element ref="ipmpmsg:IPMP_Message" minOccurs="0" maxOccurs="unbounded"/>`<br>`            </sequence>`<br>`        </extension>`<br>`    </complexContent>`<br>`</complexType>` |

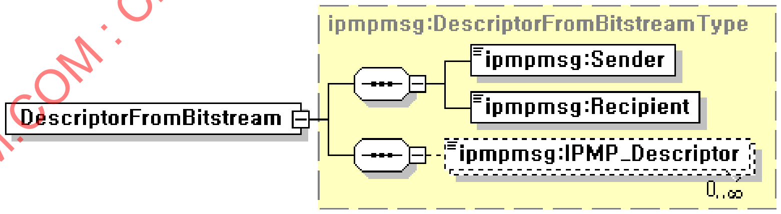
### 6.4.3 IPMP_Message

The IPMP_Message is defined below:

| | |
|---|---|
| **Diagram** |   ipmpmsg:IPMP_MessageType<br>IPMP_Message<br>ipmpmsg:URLString<br>ipmpmsg:IPMP_DescriptorIDEx<br>ipmpmsg:Data_BaseClass<br>0..∞<br>ipmpmsg:IPMP_data |
| **Children** | <URLString> <IPMP_Data> <choice>[<IPMP_DescriptorIDEx>, <Data_BaseClass>] |
| **Source** | <element name="IPMP_Message" type="ipmpmsg:IPMP_MessageType"/><br><complexType name="IPMP_MessageType"><br>    <complexContent><br>        <extension base="ipmpmsg:IPMPBaseType"><br>            <sequence><br>                <choice><br>                    <element name="URLString" type="anyURI"/><br>                    <sequence><br>                        <element name="IPMP_DescriptorIDEx" type="anyURI"/><br>                        <element ref="ipmpmsg:Data_BaseClass" minOccurs="0"<br>maxOccurs="unbounded"/><br>                    </sequence><br>                    <element name="IPMP_data" type="hexBinary"/><br>                </choice><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

### 6.4.4 DescriptorFromBitstream

The binary version of this message is used to convey binary IPMP Descriptors to an IPMP Tool.

| | |
|---|---|
| **Diagram** |   ipmpmsg:DescriptorFromBitstreamType<br>DescriptorFromBitstream<br>ipmpmsg:Sender<br>ipmpmsg:Recipient<br>ipmpmsg:IPMP_Descriptor<br>0..∞ |
| **Children** | <Sender> <Recipient> <IPMP_Descriptor> |
| **Source** | <element name="DescriptorFromBitstream"<br>type="ipmpmsg:DescriptorFromBitstreamType"<br>substitutionGroup="ipmpmsg:ToolMessageBase"/><br><complexType name="DescriptorFromBitstreamType"><br>    <complexContent><br>        <extension base="ipmpmsg:ToolMessageBaseType"><br>            <sequence><br>                <element name="IPMP_Descriptor" type="hexBinary" minOccurs="0"<br>maxOccurs="unbounded"/><br>            </sequence><br>        </extension><br>    </complexContent><br></complexType> |

The ipmpmsg:DescriptorFromBitstream element extends the ipmpmsg:ToolMessageBaseType complex type by adding a sequence of IPMP_Descriptor elements of type hexBinary.

# Annex A
(informative)

# IPMP Tool Management Protocols

## A.1 Introduction

This annex describes how an IPMP Processor uses the XML IPMP messages to instantiate, authenticate, initialise and carry out all management tasks related to IPMP Tools. These operations are necessary for granting interoperability in a device supporting a plurality of IPMP Tools from a number of providers.

The protocols are based on concepts derived from ISO/IEC 13818-11 and ISO/IEC 14496-13.

When an IPMP Processor has to supervise IPMP operations over a resource protected by one or more IPMP Tools, it needs to locate the IPMP Tools signalled in the associated IPMP Information Descriptor. The IPMP Processor extracts the IPMP Tool IDs from the IPMP Information Descriptor and determines the location where the required IPMP Tools or the IPMP Tool Pack reside. Finally, the IPMP Processor instantiates the IPMP Tools as indicated in the ipmpinfo-msx:Instantiation_API_ID [1].

Managing an IPMP Tool involves the delivery of the IPMP Messages defined in this document. IPMP Messages are exchanged between an IPMP Processor and an IPMP Tool according to a Messaging API which is particular to each device. The IPMP Processor chooses the IPMP Tools characterized by the ipmpinfo-msx:Messaging_API_ID [1] supported by the device on which it is running.

IPMP Messages are conveyed in either ipmpmsg:MessageFromTool or ipmpmsg:MessageFromDID messages. The IPMP Tool receiving such messages will obtain from the container Message the information about the Context ID of the Sender (the “Sender” element) and its own Context ID (the “Recipient” element).

The instantiation of an IPMP Tool involves the connection of the IPMP Tool instance to a specific Control Point and in a specific order compared to other IPMP Tools operating in the same Control Point (see the "order" attribute of ipmpinfo:Tool for more information on the order of IPMP Tools in a Control Point). A Registration Authority for registering Control Points on Devices, Instantiation APIs and Messaging APIs has already been established by MPEG in the context of MPEG-2 and MPEG-4 IPMPX.

## A.2 Instantiation of an IPMP Tool

### A.2.1 Introduction

Depending on the nature of the IPMP Tool – whether this is a Single IPMP Tool or an IPMP Tool Pack - the IPMP Processor will act as specified in the following two sections.

### A.2.2 Instantiation of a single IPMP Tool

When the IPMP Tool is a Single IPMP Tool, the IPMP Processor instantiates it according to the mechanisms of the platform on which it is running. According to ipmpinfo-msx:DeviceInformation [1] associated with the IPMP Tool Body, in particular the ipmpinfo-msx:ToolAPI_Config element, the IPMP Processor knows the Instantiation_API_ID information of the IPMP Tool and acts appropriately in order to get a handle of an instance of the IPMP Tool.

### A.2.3 Instantiation of an IPMP Tool Pack

When the IPMP Tool is an IPMP Tool Pack, the IPMP Processor instantiates the IPMP Tool Agent part of the IPMP Tool Pack, according to the mechanisms of the platform on which it is running. According to the ipmpinfo-msx:DeviceInformation associated with the IPMP Tool Pack Body, in particular the ipmpinfo-msx:ToolAPI_Config element, the IPMP Processor knows the Instantiation_API_ID information of the IPMP Tool Agent and acts appropriately in order to get a handle of an instance of the IPMP Tool Agent. The instantiation of the IPMP Tool Pack at this point is not complete yet because the IPMP Processor still needs to instantiate the IPMP Tools in the IPMP Tool Group. Before these can be instantiated, the IPMP Tool Agent has to be initialised as specified below.

## A.3 Initialisation of an IPMP Tool

### A.3.1 Introduction

Once the IPMP Processor has a handle of an instance of the IPMP Tool, the IPMP Processor initialises the IPMP Tool instance as specified below, depending on whether the IPMP Tool is a Single IPMP Tool or an IPMP Tool Agent.

### A.3.2 Initialization of a single IPMP Tool

The IPMP Processor initializes a Single IPMP Tool by sending it an ipmpmsg:InitialiseTool Message. This Message conveys to the IPMP Tool the following information:

- the IPMP Processor Instance, a handle of the IPMP Processor that is needed by the Tool to send Messages back to the IPMP Processor,

- the Control Point in which the IPMP Tool will operate (optional),

- the Sequence Code of the IPMP Tool defining its order with respect to other IPMP Tools operating in the same Control Point (optional),

- any IPMP Message addressed to that IPMP Tool found in the DI (optional).

### A.3.3 Initialization of an IPMP Tool Pack

The initialization of an IPMP Tool Pack involves first the initialisation of the IPMP Tool Agent.

The IPMP Processor initializes an IPMP Tool Agent by sending it a ipmpmsg:InitialiseTool Message. This Message conveys to the IPMP Tool Agent the following information:

- the IPMP Processor Instance: a handle of the IPMP Processor that is needed by the Tool to send Messages back to the IPMP Processor. The format of this handle may depend on the particular device on which the IPMP Processor is running and such information depends on the ipmpinfo-msx:Instantiation_API_ID value in the ipmpinfo-msx:ToolAPI_Config.

- the list of all available Control Points on the device where the IPMP Tool Agent may instantiate the IPMP Tools in the Tool Group with the associated ipmpmsg:ControlPointAddress where the IPMP Tool Pack operates.

- any IPMP Message addressed to that IPMP Tool found in the digital item of which the protected resource on which the IPMP Tools shall operate is part.

After the IPMP Tool Agent is instantiated, initialised and authenticated, the IPMP Processor chooses from the available Control Points the ones in which the IPMP Tools in the IPMP Tool Group shall be instantiated.

Where the IPMP Tool Agent needs the instantiation of another IPMP Tool to operate, the IPMP Tool Agent requests the reference of the IPMP Tool Group from the IPMP Processor by sending a ipmpmsg:GetToolGroupReference message. The IPMP Processor replies by sending the required information contained in the ipmpmsg:GetToolGroupReferenceResponse, if the IPMP Processor can access it. In the case where this operation fails, the IPMP Processor replies with a ipmpmsg:NotifyToolEvent, specifying an event of type “TOOL_GROUP_NOT_FOUND”.

When the IPMP Tool Agent needs the instantiation of a Single IPMP Tool to operate, then the IPMP Tool Agent requests the reference of the Single IPMP Tool from the IPMP Processor by sending a ipmpmsg:GetToolReference message specifying a list of IPMP Tool IDs of the required IPMP Tools. In the case where the IPMP Processor can access the requested IPMP Tools, the IPMP Processor replies with a ipmpmsg:GetToolReferenceResponse conveying a list of IPMP Tool IDs and the associated references of each IPMP Tool.

Once the IPMP Tool Agent obtains the reference of the IPMP Tool Group or the reference of a Single IPMP Tool, the IPMP Tool Agent connects each IPMP Tool in the Tool Group or the Single IPMP Tool to the appropriate Control Point.

The initialization information for each IPMP Tool is contained in the Tool Pack Data. The IPMP Tool Agent may use the Tool Pack Data to initialize the IPMP Tools in the Tool Group. In this case, the IPMP Tool Agent requests the Tool Pack Data from the IPMP Processor by sending an ipmpmsg:GetToolPackData message. The IPMP Processor searches for the Tool Pack Data associated with the Tool Pack of the requesting Tool Agent and sends the ipmpmsg:ToolPackData to the IPMP Tool Agent by including it in an ipmpmsg:MessageFromDID message. Finally, the IPMP Tool Agent initialises each IPMP Tool with the Tool Pack Data.

## A.4 Authentication between IPMP Tools and the IPMP Processor

As a general practice, the IPMP Processor and the IPMP Tools shall mutually authenticate before any other action is performed in the device after IPMP Tool Initialisation. This also has the advantage of allowing the establishment of a secure channel for communication among parties.

Mutual authentication can be triggered by either an IPMP Tool or the IPMP Processor by sending to the counterpart an ipmpmsg:InitAuthentication message for negotiating the mutual authentication algorithms supported by both parties. Following, a number of ipmpmsg:MutualAuthentication messages will be exchanged between the parties involved until mutual authentication is achieved. For more information on the use of these two messages refer to ISO/IEC 13818-11 and ISO/IEC 14496-13.

When the IPMP Tool involved in the authentication process is an IPMP Tool Pack, the authentication process will be performed by the IPMP Tool Agent, as the authentication between this and the IPMP Tools in the Tool Group is performed by the Tool Agent in a proprietary fashion.

## A.5 General IPMP Tool Management

### A.5.1 Introduction

ISO/IEC 13818-11 and ISO/IEC 14496-13 describe several scenarios where IPMP Messages are employed to achieve certain goals such as the exchange critical information like licenses, decryption keys, watermarking information, etc. For more information on the use of IPMP Messages for achieving these goals, refer to these two International Standards.

### A.5.2 Obtaining a reference to the IPMP Tool Group

The IPMP Processor searches the required IPMP Tool Pack and instantiates the IPMP Tool Agent. The IPMP Tool Agent may need the IPMP Tools part of the IPMP Tool Group. In this case, the IPMP Tool Agent sends a ipmpmsg:GetToolGroupReference message to the IPMP Processor to request a reference to the IPMP Tool Group.

### A.5.3 Obtaining a reference to an IPMP Tool

The IPMP Tool Agent may also employ a Single IPMP Tool for performing IPMP Functions. In this case, the Tool Agent sends ipmpmsg:GetToolReference message to the IPMP Processor to request the reference to the Single IPMP Tool.

In response to a ipmpmsg:GetToolReference message, the IPMP Processor sends a ipmpmsg: GetToolReferenceResponse message to the IPMP Tool Agent conveying the reference of the Single IPMP Tool.

# Annex B
(informative)

# The ipmpmsg schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
       xmlns="http://www.w3.org/2001/XMLSchema"
       xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
       xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
       xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
       xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
       xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
       xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
       xmlns:ipmpdidl="urn:mpeg:mpeg21:2004:01-IPMPDIDL-NS"
       elementFormDefault="qualified" attributeFormDefault="unqualified">
   <!-- IPMPBaseType -->
   <complexType name="IPMPBaseType" abstract="true"/>
   <!--ToolMessageBase-->
   <element name="ToolMessageBase" type="ipmpmsg:ToolMessageBaseType"
abstract="true"/>
   <complexType name="ToolMessageBaseType" abstract="true">
       <complexContent>
          <extension base="ipmpmsg:IPMPBaseType">
             <sequence>
                 <element name="Sender" type="anyURI"/>
                 <element name="Recipient" type="anyURI"/>
             </sequence>
          </extension>
       </complexContent>
   </complexType>
   <!--Data_BaseClass-->
   <element name="Data_BaseClass" type="ipmpmsg:Data_BaseClassType"
abstract="true"/>
   <complexType name="Data_BaseClassType" abstract="true">
       <complexContent>
          <extension base="ipmpmsg:IPMPBaseType">
             <sequence>
                 <element name="dataID" type="unsignedInt"/>
             </sequence>
          </extension>
       </complexContent>
   </complexType>
   <!-- **************************************************************** -->
   <!--  DRM Message Containers  -->
   <!-- **************************************************************** -->
   <!-- MessageFromDI-->
   <element name="MessageFromDI" type="ipmpmsg:MessageFromDIType"
substitutionGroup="ipmpmsg:ToolMessageBase"/>
   <complexType name="MessageFromDIType">
       <complexContent>
          <extension base="ipmpmsg:ToolMessageBaseType">
             <sequence>
                 <element ref="ipmpmsg:Data_BaseClass" minOccurs="0"
maxOccurs="unbounded"/>
             </sequence>
```

```
            </extension>
        </complexContent>
    </complexType>
    <!--MessageFromTool-->
    <element name="MessageFromTool" type="ipmpmsg:MessageFromToolType"
substitutionGroup="ipmpmsg:ToolMessageBase"/>
    <complexType name="MessageFromToolType">
        <complexContent>
            <extension base="ipmpmsg:ToolMessageBaseType">
                <sequence>
                    <element ref="ipmpmsg:Data_BaseClass" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <!-- ***************************************************************** -->
    <!-- DRM Messages -->
    <!-- ***************************************************************** -->
    <!-- AUTHENTICATION MESSAGES -->
    <!--InitAuthentication-->
    <element name="InitAuthentication" type="ipmpmsg:InitAuthenticationType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="InitAuthenticationType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType">
                <sequence>
                    <element name="ContextID" type="anyURI" minOccurs="0"/>
                    <element name="AuthType" type="ipmpmsg:AUTType"/>
                    <!--Context ID of the logical instance of the Tool with which
mutual authentication is to be performed-->
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <simpleType name="AUTType">
        <annotation>
            <documentation>
                "01" - No Authentication Required
                "02" - No ID verify, Do secure channel
                "03" - Do ID verify, No secure channel
                "04" - Do ID verify, Do secure channel
            </documentation>
        </annotation>
        <restriction base="integer">
            <enumeration value="01"/>
            <enumeration value="02"/>
            <enumeration value="03"/>
            <enumeration value="04"/>
        </restriction>
    </simpleType>
    <!--MutualAuthentication-->
    <element name="MutualAuthentication" type="ipmpmsg:MutualAuthenticationType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="MutualAuthenticationType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType">
                <sequence>
                    <choice minOccurs="0">
```

```
                  <element name="requestNegotiation"
type="ipmpmsg:requestNegotiationType"/>
                  <element name="successNegotiation"
type="ipmpmsg:successNegotiationType"/>
                  <element name="failedNegotiation" type="boolean"
fixed="true"/>
               </choice>
               <element name="authenticationData" type="hexBinary"
minOccurs="0"/>
               <element name="authCodes" type="ipmpmsg:AuthCodesType"
minOccurs="0"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <complexType name="requestNegotiationType">
      <sequence>
         <element name="candidateAlgorithms"
type="ipmpmsg:AlgorithmDescriptorType"/>
      </sequence>
   </complexType>
   <complexType name="AlgorithmDescriptorType">
      <sequence>
         <element name="algoID" type="anyURI" maxOccurs="unbounded"/>
         <element name="opaqueData" type="base64Binary" minOccurs="0"/>
      </sequence>
   </complexType>
   <complexType name="successNegotiationType">
      <sequence>
         <element name="agreedAlgorithms" type="ipmpmsg:AlgorithmDescriptorType"
maxOccurs="unbounded"/>
      </sequence>
   </complexType>
   <complexType name="AuthCodesType">
      <sequence>
         <element name="certificates" type="dsig:KeyInfoType"
maxOccurs="unbounded"/>
         <element name="trustData" type="hexBinary" minOccurs="0"/>
      </sequence>
   </complexType>
   <!--SecureContainer-->
   <element name="SecureContainer" type="ipmpmsg:SecureContainerType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <annotation>
      <documentation>
         To fill the protectedMsgTag field below, use values from
         Table 1 in doc ISO/IEC 14496-13:2004
      </documentation>
   </annotation>
   <complexType name="SecureContainerType">
      <complexContent>
         <extension base="ipmpmsg:Data_BaseClassType">
            <choice>
               <sequence>
                  <element name="encryptedData" type="hexBinary"/>
                  <element name="MAC" type="hexBinary" minOccurs="0"/>
               </sequence>
               <sequence>
                  <element name="protectedMsgTag" type="short"/>
                  <element name="protectedMsg" type="hexBinary"/>
```

```
                        <element name="MAC" type="hexBinary" minOccurs="0"/>
                    </sequence>
                </choice>
            </extension>
        </complexContent>
    </complexType>
    <!-- DRM TOOL CONNECTION AND DISCONNECTION MESSAGES -->
    <!--GetTools-->
    <element name="GetTools" type="ipmpmsg:GetToolsType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="GetToolsType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType"/>
        </complexContent>
    </complexType>
    <!--GetToolsResponse-->
    <element name="GetToolsResponse" type="ipmpmsg:GetToolsResponseType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="GetToolsResponseType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType">
                <sequence>
                    <element ref="ipmpmsg:Tool" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="Tool" type="ipmpmsg:toolType"/>
    <!--It represents a DRM Tool, an extension of ipmpinfo:Tool-->
    <complexType name="toolType">
        <complexContent>
            <extension base="ipmpinfo:ToolType">
                <sequence>
                    <element name="alternates" type="ipmpinfo:ToolType"
minOccurs="0"/>
                    <element ref="ipmpmsg:ParametricDescription" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <!--GetToolContext-->
    <element name="GetToolContext" type="ipmpmsg:GetToolContextType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="GetToolContextType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType"/>
        </complexContent>
    </complexType>
    <!--GetToolContextResponse-->
    <element name="GetToolContextResponse"
type="ipmpmsg:GetToolContextResponseType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
    <complexType name="GetToolContextResponseType">
        <complexContent>
            <extension base="ipmpmsg:Data_BaseClassType">
                <sequence>
                    <element name="ToolContextID" type="unsignedInt" minOccurs="0"
maxOccurs="unbounded"/>
```

```
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!--ConnectTool-->
   <element name="ConnectTool" type="ipmpmsg:ConnectToolType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <complexType name="ConnectToolType">
      <complexContent>
         <extension base="ipmpmsg:Data_BaseClassType">
            <sequence>
               <element ref="ipmpinfo:Tool"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!--DisconnectTool-->
   <element name="DisconnectTool" type="ipmpmsg:DisconnectToolType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <complexType name="DisconnectToolType">
      <complexContent>
         <extension base="ipmpmsg:Data_BaseClassType">
            <sequence>
               <element name="ToolContextID" type="unsignedInt"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!--ParamtericDescription-->
   <element name="ParametricDescription"
type="ipmpmsg:ParametricDescriptionType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <complexType name="ParametricDescriptionType">
      <complexContent>
         <extension base="ipmpmsg:Data_BaseClassType">
            <sequence>
               <element name="descriptionComment" type="string" minOccurs="0"/>
               <element name="majorVersion" type="byte"/>
               <element name="minorVersion" type="byte"/>
               <element name="paramToolDescription"
type="ipmpmsg:paramToolDescriptionType" minOccurs="0"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <complexType name="paramToolDescriptionType">
      <sequence>
         <element name="class" type="string"/>
         <element name="subClass" type="string"/>
         <element name="typeData" type="string" minOccurs="0"/>
         <element name="type" type="string" minOccurs="0"/>
         <element name="addedData" type="string" minOccurs="0"/>
      </sequence>
   </complexType>
   <!--ToolParamCapabilitiesQuery-->
   <element name="ToolParamCapabilitiesQuery"
type="ipmpmsg:ToolParamCapabilitiesQueryType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <complexType name="ToolParamCapabilitiesQueryType">
      <complexContent>
```

```
         <extension base="ipmpmsg:Data_BaseClassType">
            <sequence>
               <element name="toolParamDesc"
type="ipmpmsg:ParametricDescriptionType"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!--ToolParamCapabilitiesResponse-->
   <element name="ToolParamCapabilitiesResponse"
type="ipmpmsg:ToolParamCapabilitiesResponseType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <complexType name="ToolParamCapabilitiesResponseType">
      <complexContent>
         <extension base="ipmpmsg:Data_BaseClassType">
            <sequence>
               <element name="capabilitiesSupported" type="boolean"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- DRM TOOL NOTIFICATION -->
   <!--NotifyToolEvent-->
   <element name="NotifyToolEvent" type="ipmpmsg:NotifyToolEventType"
substitutionGroup="ipmpmsg:Data_BaseClass"/>
   <complexType name="NotifyToolEventType">
      <complexContent>
         <extension base="ipmpmsg:Data_BaseClassType">
            <sequence>
               <sequence minOccurs="0">
                  <element name="OD_ID" type="unsignedInt"/>
                  <element name="ESD_ID" type="unsignedInt"/>
               </sequence>
               <element ref="ipmpmsg:EventType" maxOccurs="unbounded"/>
               <element name="toolContextID" type="unsignedInt"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <element name="EventType" type="ipmpmsg:EvType"/>
   <simpleType name="EvType">
      <annotation>
         <documentation>
            "00" - CONNECTED
            "01" - CONNECTION_FAILED
            "02" - DISCONNECTED
            "03" - DISCONNECTION_FAILED
            "04" - WATERMARKDETECTED
            "05" - PARSE_TOOLPACKDATA_SUCCESS
            "06" - PARSE_TOOLPACKDATA_FAILED
            "07" - UNABLE_TO_PROCESS
            "08" - TOOL_GROUP_NOT_FOUND
               "09" - TERMINATION_FAILED
             "10" - CONTROLPOINT_NOT_SUPPORTED
             "11" - UNABLE_TO_PARSE_LICENSE
             "12" - NO_VALID_LICENSE
             "13" - LICENSE_VALIDATION_FAILED
             "14" - READY_TO_PLAY
             "15" - READY_TO_BE_TERMINATED
         </documentation>
```