INTERNATIONAL STANDARD

ISO/IEC 14496-1

Second edition
2001-10-01

AMENDMENT 3
2004-05-01

Information technology — Coding of audio-visual objects —

Part 1: **Systems**

AMENDMENT 3: Intellectual Property Management and Protection (IPMP) extensions

Technologies de l'information — Codage des objets audiovisuels — Partie 1: Systèmes

AMENDEMENT 3: Gestion et extensions de protection de la propriété intellectuelle (IPMP)

Reference number
ISO/IEC 14496-1:2001/Amd.3:2004(E)

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

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.

Amendment 3 to ISO/IEC 14496-1:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Introduction

This document specifies IPMP extensions to the currently specified IPMP specification (the “Hooks”), documented in ISO/IEC 14496-1:2001.

“Hooks” Terminals do not support the additional functionality specified here. Hence, “Hooks” Terminals will not be able to fully process “Extensions” content. However, the syntax and semantics here are compliant with the “Hooks” framework, and therefore allow for graceful failure of a well-designed Terminals in either the event that they are compliant to non-extended IPMP only and receive extended IPMP conforming content, that they are compliant to extended IPMP as outlined in this document and able to process either non-extended conforming content or extended conforming content to the extent that their implementations allow.

Additionally, although this amendment does not preclude the use of “Hooks” and “Extensions” tools being used in the same MPEG-4 presentation, the behaviour of both “Hooks” as well as “Extensions” being used for the protection of the same stream is undefined.

This Amendment specifies

- Extensions to the OD framework in a backward compatible manner to support the use of IPMP in the protection of OD and scene description streams.
- Extensions to the OD framework to support the Identification of required IPMP tools using either 128 bit registered Ids or parametric descriptions.
- Extensions to `IPMP_DescriptorPointer` in a backward compatible manner to support the extended addressing of IPMP streams.
- Extensions to `IPMP_Descriptor` in a backward compatible manner to support the use of 16 bit identifiers as well as supporting the identification of the location within a given stream where the specified IPMP tool is to be placed as well as supporting the sequencing of multiple tools at the same location.
- Extensions to `IPMP_Descriptor` in a backward compatible manner to support the carriage of normative IPMP information.
- Extensions to `IPMP_Message` in a backward compatible manner to support the specific addressing of a given `IPMP_Message` to specific IPMP tools.
- Extensions to `IPMP_Message` in a backward compatible manner to support the carriage of normative IPMP information.
- The definition, as well as Extension tags, syntax and semantics for an `IPMP_Data_BaseClass` to support the following functionalities.
  - ✧ Mutual Authentication for IPMP tool to IPMP tool as well as IPMP tool to Terminal communication.
  - ✧ The requesting by IPMP tools of the connection/disconnection to requested IPMP tools.
  - ✧ The notification to IPMP tools of the connection/disconnection of IPMP tools.
  - ✧ Common IPMP processing.
  - ✧ IPMP tool to/from User interaction.

- Syntax and semantics for the carriage of IPMP tools in the bit stream.
- Syntax and semantics for IPMP information carriage to and from IPMP tools.
- Syntax and semantics for the requesting and transfer of content and IPMP Tools between Terminals as well as extension tags, syntax and semantics to the IPMP_Data_BaseClass ISO/IEC 14496-1 used therein.
- XML syntax and semantics for the description of the environment in which and MPEG-4 Terminal/application is operating.
- A list of registration authorities required for the support of the amended specifications found herein.

Information technology — Coding of audio-visual objects —

Part 1:
Systems

AMENDMENT 3: Intellectual Property Management and Protection (IPMP) extensions

In Clause 0.6.1.1, replace the text:

*"*The intellectual property management and protection (IPMP) framework for ISO/IEC 14496 content consists of a normative interface that permits an ISO/IEC 14496 terminal to host one or more IPMP Systems. The IPMP
interface consists of IPMP elementary streams and IPMP descriptors. IPMP descriptors are carried as part of an
object descriptor stream. IPMP elementary streams carry time variant IPMP information that can be associated to multiple object descriptors.
The IPMP System itself is a non-normative component that provides intellectual property management and protection functions for the terminal. The IPMP System uses the information carried by the IPMP elementary streams and descriptors to make protected ISO/IEC 14496 content available to the terminal. An application may
choose not to use an IPMP System, thereby offering no management and protection features."

with:

"The intellectual property management and protection (IPMP) framework for ISO/IEC 14496 content consists of a normative interface that permits an ISO/IEC 14496 terminal to host one or more IPMP Systems in the form of monolithic IPMP Systems or modular IPMP Tools. The IPMP interface consists of IPMP elementary streams and IPMP descriptors. IPMP descriptors are carried as part of an object descriptor stream. IPMP elementary streams carry time variant IPMP information that can be associated to multiple object descriptors. The IPMP System, or IPMP Tools themselves are non-normative components that provides intellectual property management and protection functions for the terminal. The IPMP Systems or Tools uses the information carried by the IPMP elementary streams and descriptors to make protected ISO/IEC 14496 content available to the terminal. An application may choose not to use an IPMP System, thereby offering no management and protection features."

In Clause 4.38, replace the text:

"Only the interface to such systems is normatively defined."

with the text:

"The interface to such systems is defined as well as :

- The provision for the identification of IPMP Tools either through the use of a registration authority or through the use of a functional description of the IPMP Tools' capabilities in a parametric fashion.
- Controlling the time of instantiation of IPMP Tools either by the inclusion of references to the required IPMP Tools or at the request of already instantiated IPMP Tools.

- Providing secure messaging between IPMP Tools and the Terminal and between IPMP Tools and the User.
- Notification of the instantiation of IPMP Tools to IPMP Tools requesting such notification.
- Interaction between IPMP Tools, and/or the Terminal and the User.
- The carriage of IPMP Tools within the bitstream."

In Clause 4, insert the following text alphabetically:

Binary Representation

In the context of an IPMP Tool, this is the format of the implementation of that IPMP Tool, Examples: Platform Dependent Native Code, Java ™ bytecode.

Content

This implies part or whole of an MPEG presentation.

Content Consumption

Any experience of given Content implies consumption of that content. Access, Playback, Denial of Access and Creation of a Copy are all types of content consumption.

Content Stream

This is the incoming content, of MPEG-4 format.

Control Point

A point on a given elementary stream in a Terminal where IPMP Processing on stream data shall be carried out.

IPMP Device

An implemented application that implements an MPEG-4 Terminal supporting the use of MPEG-4 IPMP.

IPMP Information

Information directed to a given IPMP Tool to enable, assist or facilitate its operation.

IPMP System

A monolithic IPMP protection scheme which requires implementation dependant access to protected streams at required Control Points and must provide any intra-communication within an IPMP System on an implementation basis.

IPMP Tool

IPMP tools are modules that perform (one or more) IPMP functions such as authentication, decryption, watermarking, etc. Conceptually the use of one or more IPMP Tools is combined to perform the functionality of an IPMP System. IPMP Tools, as opposed to IPMP Systems, are normatively identified as to which control points they function at as well as are provided normative methods for secure communications both within as well as outside of a given IPMP Tools comprised functional "IPMP System". An additional difference between IPMP Tools and IPMP Systems is that IPMP Tools, or a combination thereof, may be used for the protection of Object streams.

In this amendment the use of the term “IPMP System” is used in some cases to indicate either an actual IPMP System or a combination of IPMP Tools whose combination provides the functionality of an IPMP System. In cases where the distinction is important the proper respective terms are used.

IPMP Tool Identifier

This refers to the IPMP Tool ID. It identifies a Tool in an unambiguous way, at the presentation level or at a universal level. Two different identifiers are provided to support the differentiation between the use of IPMP Systems and IPMP Tools.

IPMP Tool List

The IPMP Tool List identifies, and enables selection of, the IPMP Tools required to process the Content.

IPMP Tool Manager

The IPMP Tool Manager is a conceptual entity within the Terminal that processes IPMP Tool List(s) and retrieves the Tools that are specified therein.

IPMP Tool Message

A message passed between any combination of IPMP Tool or Terminal.

IPMP Tool Stream

An elementary stream carrying an implementation of an IPMP Tool.

Message Router

A conceptual entity within the Terminal that implements the Terminal-side behavior of the Terminal-Tool interface.

Mutual Authentication

Protocols carried out to determine the proper and correct identity of a communicating entity and to secure the communication channels between communicating entities.

Parametric Configuration

Information that carries task-specific parameter specification in an extensible form.

Parametric Description

Parametrically described tools shall be defined by an SDL declaration that governs a given description, the parametric configuration and other interface message(s) that drive the tool and the behaviour defined for fulfilment of such a description.

Representation Format

The binary format, platform and communication mechanisms applicable to a given implementation of an IPMP Tool or Terminal.

Scope of Protection

Scope of protection refers to the elementary stream and/or object governed by a given IPMP Tool instance.

Terminal

A Terminal is an environment that consumes possibly protected Content in compliance with the usage rules.

User

A hardware, software or human entity that is the initiator and/or target of content consumption.

In Subclause 8.2.1, replace Table 1 with the following:

Table 1 - List of Class Tags for Descriptors

Tag value	Tag name
0x00	Forbidden
0x01	ObjectDescrTag
0x02	InitialObjectDescrTag
0x03	ES_DescrTag
0x04	DecoderConfigDescrTag
0x05	DecSpecificInfoTag
0x06	SLConfigDescrTag
0x07	ContentIdentDescrTag
0x08	SupplContentIdentDescrTag
0x09	IPI_DescrPointerTag
0x0A	IPMP_DescrPointerTag
0x0B	IPMP_DescrTag
0x0C	QoS_DescrTag
0x0D	RegistrationDescrTag
0x0E	ES_ID_IncTag
0x0F	ES_ID_RefTag
0x10	MP4_IOD_Tag
0x11	MP4_OD_Tag
0x12	IPL_DescrPointerRefTag
0x13	ExtendedProfileLevelDescrTag
0x14	profileLevelIndicationIndexDescrTag
0x15-0x3F	Reserved for ISO use
0x40	ContentClassificationDescrTag
0x41	KeyWordDescrTag
0x42	RatingDescrTag
0x43	LanguageDescrTag
0x44	ShortTextualDescrTag
0x45	ExpandedTextualDescrTag
0x46	ContentCreatorNameDescrTag
0x47	ContentCreationDateDescrTag
0x48	OCICreatorNameDescrTag
0x49	OCICreationDateDescrTag
0x4A	SmpteCameraPositionDescrTag
0x4B-0x5F	Reserved for ISO use (OCI extensions)
0x60	IPMP_ToolsListDescrTag
0x61	IPMP_ToolTag
0x62-0xBF	Reserved for ISO use
0xC0-0xFE	User private
0xFF	Forbidden

In Clause 8.3, replace the title:

Intellectual Property Management and Protection (IPMP)

with:

Intellectual Property Management and Protection framework (IPMP)

In Subclause 8.3.1, replace the entire text with:

The intellectual property management and protection (IPMP) framework for ISO/IEC 14496 content consists of a normative interface that permits an ISO/IEC 14496 terminal to host one or more IPMP Systems or IPMP Tools. Additionally, the framework contains a secure messaging system usable between IPMP Tools as well as IPMP Tools and the Terminal and IPMP Tools and the User.
An IPMP System or IPMP Tools are non-normative components that provide intellectual property management and protection functions for the terminal.
The IPMP interface consists of IPMP elementary streams and IPMP descriptors. The normative structure of IPMP elementary streams is specified in this Subclause. IPMP descriptors are carried as part of an object descriptor stream and are specified in 8.6.14. The IPMP interface allows applications (or derivative application standards) to build specialized IPMP Systems or IPMP Tools. Alternatively, an application may choose not to use an IPMP System or IPMP Tools, thereby offering no management and protection features. The IPMP System and IPMP Tools use the information carried by the IPMP elementary streams and descriptors to make protected ISO/IEC 14496 content available to the terminal. The detailed semantics and decoding process of the IPMP System or IPMP Tools are not in the scope of ISO/IEC 14496. The usage of the IPMP System/Tools Interface, however, is explained in 8.8, where the usage of the IPMP framework is also explained.

8.3.1.1 IPMP Architecture overview

This clause describes the general IPMP architecture. For detailed IPMP architecture linked to MPEG-4 system, please refer to 8.8.6.

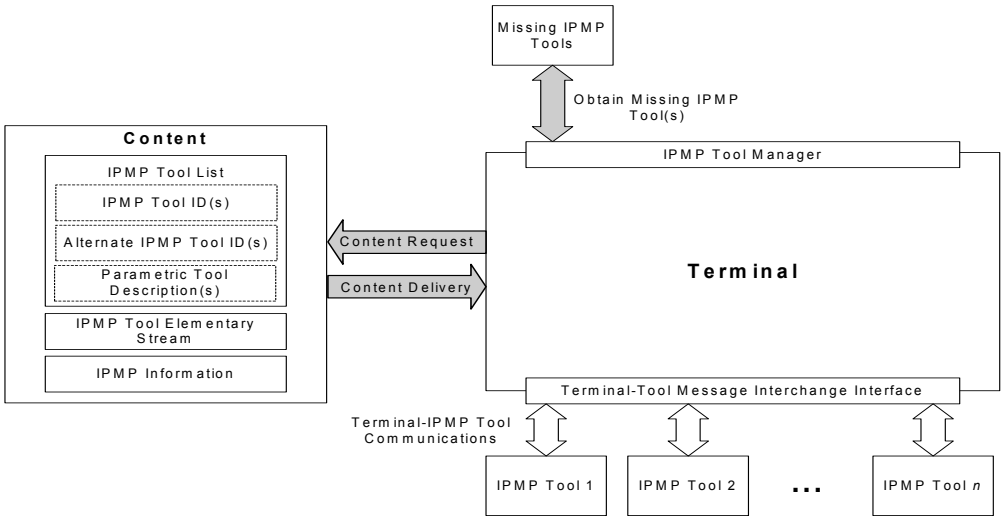

Figure AMD3-1 — Architecture Diagram for General Concepts

8.3.1.1.1 Messaging

To facilitate the cooperation of multiple tools in the protection and governance of content, a message based architecture is provided. The message based architecture has three advantages over functional interface type architectures. The first is that security can more easily be maintained as messages are less difficult to protect in an open framework then parameters in a function parameters list. The second is that the only entities that need be concerned with a given message's definition are those that need to generate or act upon a given message and so additional functionality can be created and supported simply through the addition of required messages. The third is that full interoperability with IPMP tools can be easily achieved by using the `IPMP_ToolAPI_Config` [8.3.2.12.5] carried in IPMP Descriptor, or by defining a single messaging API by a third-party forum who adopts IPMP.

Physical routing of information and context resolution are handled by a conceptual entity called the Message Router. The Message Router abstracts all platform-dependent routing and delivery issues, from the IPMP Tools. The interface between the Message Router and the Tools, is non-normative and is not defined in this specification, however, the information on the messaging interface can be carried in `IPMP_ToolAPI_Config` to assist interoperability.

All IPMP Tool interactions take place via the Terminal. IPMP Tools do not communicate directly with each other within the scope of this standard.

The delivery of both bit stream sourced IPMP information as well as IPMP tool and Terminal generated information is supported through the use of three separate messages which are passed via the Message Router to IPMP tools. The three messages are, `IPMP_MessageFromBitstream` [8.3.2.12.2], which is used to deliver IPMP stream data, `IPMP_DescriptorFromBitstream` [8.3.2.12.3], which is used to deliver `IPMP_Descriptors` [ISO/IEC 14496-1] and `IPMP_MessageFromTool` [8.3.2.12.4], which is used to deliver messages from either other IPMP tools or the Terminal itself.

In these extensions, a core set of messages are provided which cover what are identified as core functional requirements.

8.3.1.1.2 Mutual Authentication

The most important aspect of a secure messaging architecture is the use of cryptographic algorithms and protocols that allow one to perform a number of important security functions.

At any point in IPMP Information or Content processing, IPMP Tools may be required to communicate with one another or the Terminal. The degree of security required for such communication is determined by a number of variables including information that may be included by the content provider in the Content and conditions of trust established between tool providers a priori and out of band. It is generally the case that a given ES is protected by multiple tools but that certain types of tools are complex (e.g. Rights Management tools) and others are utilities (e.g. Decryption engines). Complex tools may control the instantiation of other tools or make decisions about content use in response to usage queries from the terminal. Mutual authentication may occur between any pair of tools but the level of security required for this communication will in part be dictated by data contained in the bitstream in an opaque manner. The mechanism for making the determination of this security level is non-normative.

Mutual authentication is executed as follows:

1. The Tool that initiates mutual authentication with another tool determines the conditions of trust to be achieved by such authentication, i.e. the initiating tool determines whether it needs integrity protected communication or full secure, authenticated communication. This level may or may not be dictated by IPMP Information in the Content.

2. The communicating tools then engage in a message exchange to determine which authentication protocol will be used. In some cases, this protocol will have been determined by an a priori out of band negotiation between the tool providers in their security audits of one another.

These extensions provide a set of messages to support the identified functionalities. The first message `IPMP_InitAuthentication` [8.3.2.7.1] may be used and delivered to a given IPMP tool such that the receiving IPMP tool is informed as to its required communication partner as well as security measures that must be in place. Following this message or the absence thereof, IPMP tools required to do so will use the `IPMP_MutualAuthentication` [8.3.2.7.2] message as required to determine or create secure channels of communication as needed based on the application.

As one purpose of Mutual Authentication is the verification of trust relationships existing between two entities these specifications provide for the carriage of trust and security metadata. This metadata may include zero or more certificates, credentials or integrity verification information. The creation or establishment of trust relationships are established by out of band relationships between the different entities involved in protecting and managing the content. However, the trust metadata that results from such relationships needs to be made available to permit static and dynamic verification of trust.

During the Mutual Authentication process the carriage of `IPMP_TrustSecurityMetadata` [8.3.2.7.2.7] is supported to provide additional security related data usable by IPMP tools to determine trust related information.

Once Mutual Authentication is performed, the `IPMP_SecureContainer` [8.3.2.7.3] may be used to pass information securely between IPMP tools and IPMP tools and Terminal.

8.3.1.1.3 IPMP tool acquisition

ISO/IEC 14496-1 defines `IPMP_ToolListDescriptor` [8.6.14.2] which conveys the list of IPMP tools required to access the content associated with the `InitialObjectDescriptor` in which it is described, and may include a list of alternate IPMP tools or parametric descriptions of tools required to access the content. The conceptual entity Tool Manager parses the tool list, makes sure all tools are available, and retrieves missing tools if any.

If a given required tool is not present on a given Terminal, the `IPMPToolES_DecoderConfig` [8.3.2.8.3] descriptor can be used to indicate an IPMP tool bearing stream with `IPMP_ToolES_AU` [8.3.2.8.4] being used to actually carry the required tool.

A missing IPMP Tool can also be acquired from a neighbouring IPMP device via tool transfer messages defined in Annex D, or it can be acquired by connecting to a remote server and providing the terminal description as defined in Annex E.

8.3.1.1.4 IPMP Tool connection and disconnection

In the IPMP architecture, IPMP tool may be connected as the result of an `IPMP_DescriptorPointer` [ISO/IEC 14496-1] being processed and in addition may be connected due to requests from already connected IPMP tools. Note: Instantiation of the Tools to be connected is implementation dependent, however, the information on how to instantiate the Tools can be carried in `IPMP_ToolAPI_Config` [8.3.2.12.5] to assist interoperability.

IPMP tools may use the `IPMP_GetTools` [8.3.2.8.1] and `IPMP_GetToolsResponse` [8.3.2.8.2] messages to request a list of tools available for connection that exist as well as a response to the request, respectively.

IPMP tools as well as the Terminal may also query a given IPMP tool as to its capabilities and functionality by using the `IPMP_ToolParamCapabilitiesQuery` [8.3.2.8.5] message with the tool being queried using the `IPMP_ToolParamCapabilitiesResponse` [8.3.2.8.6]message as a reply.

Knowing that a given tool is needed for processing, an IPMP tool may request the connection of another IPMP tool by using the `IPMP_ConnectTool` [8.3.2.8.7] message and may request the disconnection of another IPMP tool by using the `IPMP_DisconnectTool` [8.3.2.8.8] message. A connection may require the actual instantiation of a tool or may be accomplished through physical/electronic means.

The `IPMP_ConnectTool` message contains control point and sequence information to determine the exact location to connect the requested tool. Tools connected at the request of other tools inherit the same scope of protection as the requesting tool. Note that some control points are not associated with any known point on an Elementary Stream. Note also that there are issues with scoping and scenarios that are somewhat illogical, especially as related to BIFS nodes referencing ODs.

Each instantiation of an IPMP Tool shall establish a new logical instance of the tool, for a particular scope of protection. The Terminal assigns a context identifier for the logical instance of the tool, which maps to the specific tool instance, and therefore to the associated scope of protection. These context identifiers shall be unique to ensure unambiguous message addressing.

The process of instantiation involves the following steps

1. Establish a context for the Tool being instantiated
2. Establish a link between the MR and the Tool instance
3. Establish a link between the Tool instance and the MR.

Details of this process are implementation specific. The normative result is a context/address being made available for communication and use of the instantiated tool by other tools as well as the Terminal.

The tool requesting connection shall receive an `IPMP_NotifyToolEvent` [8.3.2.9.3] message indicating the instantiation of the IPMP Tool, and its associated context. The requesting tool and the instantiated tool may perform mutual authentication thereafter.

If an IPMP tool knows of another tool with which it must communicate has already been connected, it may use the `IPMP_GetToolContext` [8.3.2.8.9] message and will receive an `IPMP_GetToolContextResponse` [8.3.2.8.10] message in reply if the requested IPMP tool is already connected with the message containing the address through which the requesting tool may use to communicate with the requested tool.

8.3.1.1.5 Notification of IPMP Tool connection and disconnection

During the processing of IPMP protected content, a number of IPMP tools may be involved, for communication and various security purposes, notification messages are supplied to notify IPMP tools when other IPMP tools have either been connected, disconnected or processed watermark information. Additionally IPMP tools may request at any time a list of all the IPMP tools currently connected at various specifiable scopes of protection.

The IPMP_AddToolNotificationListener [8.3.2.9.1] message may be used to indicate the sending IPMP tools intent to receive notifications of events and scopes of protection as specified in the IPMP_AddToolNotificationListener message. To remove one's self from being notified in the future for specified events, an IPMP tool may use the IPMP_RemoveToolNotificationListener [8.3.2.9.2] to do so.

As events occur for which notifications have been requested, the IPMP_NotifyToolEvent [8.3.2.9.3] message is sent to requesting IPMP tools.

8.3.1.1.6 Common IPMP processing

Direct support has been provided for the carrying out of common IPMP processing operations. Specific support and the related messages are as follows:

- IPMP_CanProcess [8.3.2.10.1] for the notification by an IPMP tool to the Terminal that stream processing may begin. All tools connected and processing data within a given stream must send permission before any tools in the stream receive stream data.
- IPMP_Opaquedata [8.3.2.10.2] for the carriage of user defined data.
- IPMP_KeyData [8.3.2.10.3] for the carriage of decryption key data as well as timing information to determine the validity period of time varying keys.
- IPMP_RightsData [8.3.2.10.4] for the carriage of rights expressions.
- IPMP_SelectiveDecryptionInit [**Error! Reference source not found.**] for the configuration of a decryption tool.
- IPMP_AudioWatermarkingInit [**Error! Reference source not found.**] for the configuration of an audio watermarking tool.
- IPMP_SendAudioWatermark [**Error! Reference source not found.**] for the sending of information to be embedded or that was extracted.
- IPMP_VideoWatermarkingInit [**Error! Reference source not found.**] for the configuration of a video watermarking tool.
- IPMP_SendVideoWatermark [**Error! Reference source not found.**] for the sending of information to be embedded or that was extracted.

8.3.1.1.7 IPMP tool to/from User interaction

During IPMP processing, direct interaction between IPMP tools and a User may be required. The IPMP_UserQuery [8.3.2.11.1] message is used to provide information to be relayed to a User and to request information as well. The IPMP_UserQueryResponse [8.3.2.11.2] is used to relay information provided by a User back to the originator of the User query."

In Subclause 8.3.2.5.1, replace the syntax with the following:

```
aligned(8) expandable(2^28-1) class IPMP_Message
{
   bit(16)  IPMPS_Type;
   if (IPMPS_Type == 0)
   (
      bit(8) URLString[sizeOfInstance-2];
   )
```