SMPTE RDD 24:2013

SMPTE REGISTERED DISCLOSURE DOCUMENT

Specification of the FIMS Media SOA Framework

The attached document is a Registered Disclosure Document prepared by the proponent(s) identified below. It has been examined by the appropriate SMPTE Technology Committee and is believed to contain adequate information to satisfy the objectives defined in the Scope, and to be technically consistent.

This document is NOT a Standard, Recommended Practice or Engineering Guideline, and does NOT imply a finding or representation of the Society.

Errors in this document should be reported to the proponents identified below, with a copy to eng@smpte.org. All other inquiries in respect of this document, including inquiries as to intellectual property requirements that may be attached to use of the disclosed technology, should be addressed to the proponents identified below.

Proponents contact information:

FIMS Administrative Group
Email: fims_adm@list.ebu.ch
Website: www.fims.tv

Jean-Pierre Evain
European Broadcasting Union
L'Ancienne-Route 17A
CH-1218 Grand-Saconnex Suisse
Email: evain@ebu.ch

Brad Gilmer
Advanced Media Workflow Association, Inc.
436 N. Westfield Road
Madison, WI 53717 USA
Email: brad_gilmer@amwa.tv

Approved
November 7, 2013

EBU

OPERATING EUROVISION

TECH 3356

SPECIFICATION OF THE FIMS MEDIA SOA FRAMEWORK

Part 1: General Description

VERSION 1.0.7

Published Jointly With FIMS and AMWA

Framework for Interoperable Media Services

ADVANCED MEDIA WORKFLOW ASSOCIATION

Geneva
September 2012

[*] Page intentionally left blank. This document is paginated for two sided printing

Executive Summary

This document describes a vendor-neutral common framework for implementing Interoperable Media Services using a Service Oriented Architecture (SOA) based system, supporting interoperability, interchangeability and reusability of media specific services.

The FIMS 1.0 release at the time of publication of this document comprises the following:

- FIMS 1.0 Part 1: General Description (this document):
  Part 01-General Description-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S0: Base Schema
  Part 02_S0-Base Schema-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S1: Transfer Service
  Part 02_S1-Transfer Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S2: Transform Service
  Part 02_S2-Transform Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S3: Capture Service:
  Part 02_S3-Capture Service-FIMS Schema Spec-1.0.7.pdf
- Schema file package:
  FIMS_1_0_7.zip

NOTES – The user's attention is called to the possibility that implementation and compliance with this specification may require use of subject matter covered by patent rights. By publication of this specification, no position is taken with respect to the existence or validity of any claim or of any patent rights in connection therewith. The AMWA, including the AMWA Board of Directors, shall not be responsible for identifying patents for which a license may be required by an AMWA specification or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Page intentionally left blank. This document is paginated for two sided printing

Contents

Framework for Interoperable Media Services
FIMS Media SOA Framework 1.0

Part 1: General Description

1. Scope

This document describes a vendor-neutral common framework for implementing Interoperable Media Services using a Service Oriented Architecture (SOA) based system for use in broadcast, production, post production, media distribution, and media archive applications. The framework supports interoperability, interchangeability and reusability of media specific services.

The high-level architecture and framework is described. This framework covers the following system and management requirements: service management, awareness, behaviour and communication, content and time awareness.

This first version of the specification addresses three basic Media Services: Capture, Transform, and Transfer.

This document should be read in conjunction with the following:

- FIMS 1.0 Part 2, S0: Base Schema
- FIMS 1.0 Part 2, S1: Transfer Service
- FIMS 1.0 Part 2, S2: Transform Service
- FIMS 1.0 Part 2, S3: Capture Service
- FIMS 1.0 file package

The standard major, minor and revision numerical versioning system is used for release management of the specification. 1.0.x releases of the specification shall contain minor revisions addressing inconsistencies and defects within the basic 1.0 release. Additional features to the 1.0 release will only be added to a future 1.x document.

2. Conformance Language

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly

labelled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3. Reference Documents

3.1 FIMS

1) FIMS 1.0 schema files; current file package at the time of release of this document: FIMS_V1_0_7 (September 2, 2012).
2) FIMS Part 2: Service Interfaces – S0: Base Schema
3) FIMS Part 2: Service Interfaces – S1: Transfer Service
4) FIMS Part 2: Service Interfaces – S2: Transform Service
5) FIMS Part 2: Service Interfaces – S3: Capture Service

N.B. See http://wiki.amwa.tv/ebu/ for latest versions of files and documentation.

3.2 General References

6) World Wide Web Consortium (W3C), http://www.w3.org
7) Organization for the Advancement of Structured Information Standards (OASIS), http://www.oasis-open.org
8) Web Services Interoperability Organisation (WS-I), http://www.ws-i.org
9) Advanced Media Workflow Association (AMWA), http://www.amwa.tv
10) Society of Motion Picture and Television Engineers (SMPTE), http://www.smpte.org
11) European Broadcast Union, EBU Technical References, http://www.ebu.ch

3.3 Standards References

Implementers should have a general understanding of the following technologies:

3.3.1 Web and Internet Technology

12) RFC 1738: Uniform Resource Locators (URL)

13) RFC 1521: Mechanisms for Specifying and Describing the Format of Internet Message Bodies

3.3.2 SOAP Web Services

14) WSDL 1.1

15) Web Services Description Language (WSDL) Version 2.0 Part 0: Primer - W3C Recommendation 26 June 2007. Available at http://www.w3.org/TR/wsdl20-primer.

3.3.3 Date and time formats

16) EBU Tech 3295 P_META Metadata Library

17) ISO 8601:2004 Data elements and interchange formats -- Information interchange -- Representation of dates and times

18) IETF RFC 3339 Date and Time on the Internet: Timestamps

3.3.4 EBU studies/specifications and SMPTE Standards

19) EBU Tech 3293: EBU Core Metadata Set (EBUCore)

20) EBU Tech 3295: P_META Metadata Library

21) SMPTE ST 12M-1:2008 Television – Time and Control Code

22) SMPTE RP 224v11:2011: SMPTE Labels Register

23) SMPTE ST 258:2004 Television - Transfer of Edit Decision Lists

24) SMPTE ST 291:2010 Television - Ancillary Data Packet and Space Formatting

25) SMPTE ST 377-1:2009 Material Exchange Format (MXF) – File Format Specification

26) SMPTE ST 330:2004 Television - Unique Material Identifier (UMID)

27) SMPTE ST 434: 2006 Material Exchange Format - XML Encoding for Metadata and File Structure Information

28) SMPTE ST 436: 2006 Television - MXF Mappings for VBI Lines and Ancillary Data Packets

29) SMPTE ST 2032-1:2007 Media Dispatch Protocol (MDP) - Protocol Specification

4. Part 1 Overview

This section is informative.

This specification defines a common approach to the integration of software components in modern media production facilities, which is believed to be a fundamental need of the entire media industry.

The specification is based on an overall framework for integration of reusable components for multimedia content production, which would support the business functions of the professional media industry. This framework for media services includes specific detailed definitions of common media service interfaces.

The focus of this specification is on Service Oriented Architecture (SOA), and reflects a move by the media industry toward systems that use this approach. Media companies are moving towards more rapid adoption of these systems because of a need to increase agility in a market where user requirements are changing rapidly. In addition to the need for agility, these companies require the increased maintainability, scalability and extensibility that standardized service interfaces provide.

To properly exploit this technology a common framework should be adopted to help ensure integration interoperability, interchangeability and reusability of services. This will drastically reduce integration costs, allow users to more freely choose the most appropriate products on the market at any given time, improve maintainability, and aid in the adoption of new technologies.

Section 5 gives an overview of the high-level architecture, and provides the overall reference model of the FIMS Framework.

Section 6 describes media service management, and specifies Service lifecycle and the lifecycle of a job.

Section 7 describes media service awareness, including the Service registry, and Service description.

Media Service behaviour is described in **Section 8**. This includes the service behaviour that is common to all categories of services. Behaviour specific to each service category is also described.

Media Service communication is specified in **Part 2, Schema Description**. The FIMS object model is described by a set of XML schemas, which provide the object model representation for common objects and extensions for the different classes of service. Part 2 Section S0 defines the base schema, and service interfaces are specified in subsequent sections.

It is proposed that future versions of the FIMS specification will expand the number of supported media service interfaces.

5. High-Level Architectures

5.1 What does it mean to be 'FIMS Compliant'?

Compliance with FIMS 1.0 requires that the following conditions shall be met:

1) Messages shall be implemented as described in the FIMS service description framework

2) Message names or defined schemas shall not be modified.

3) Messages shall not be processed in a means that is dependent on schema extensions.

4) Each service interface shall comply with the FIMS WSDL in the case of SOAP implementations.

5) Each service shall implement at least the mandatory parts of the FIMS WSDL definition in the case of SOAP implementations, and shall not extend them except for fields that explicitly allow vendor extensions.

5.2 Reference architecture(s)

Figure 1 shows the overall reference model of the FIMS Framework, including areas not specified in FIMS 1.0.

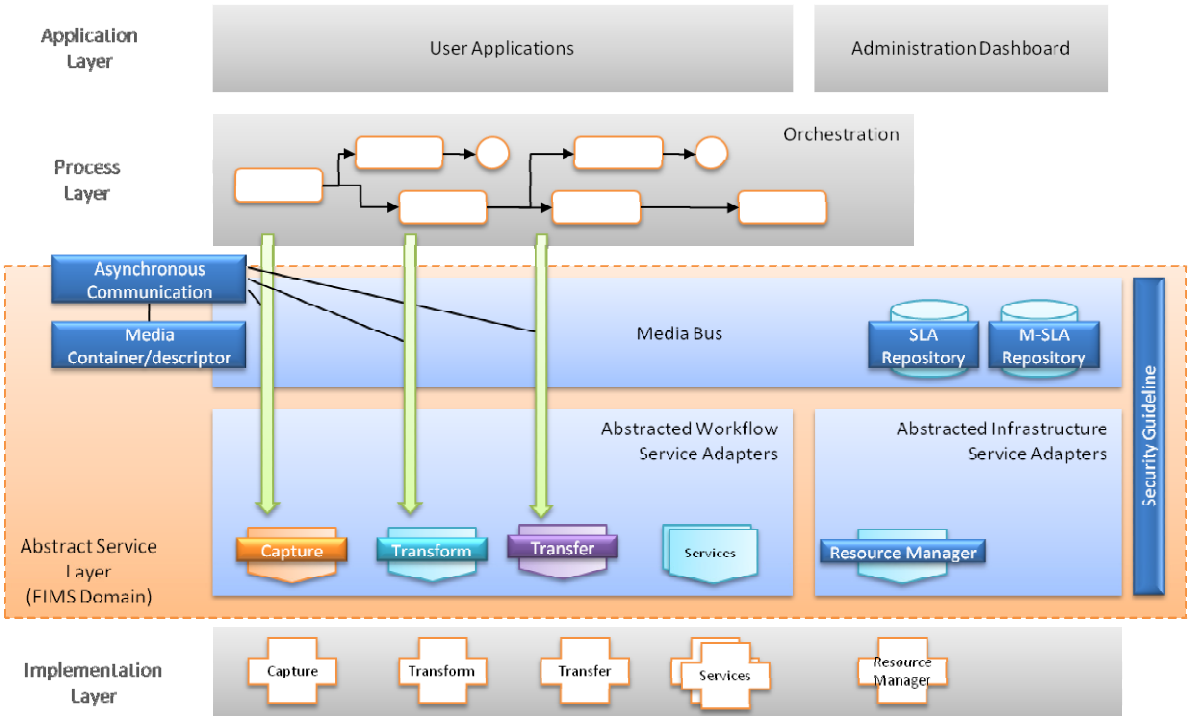


Figure 1: FIMS Framework reference model

To take account of the specific needs of audiovisual media, the FIMS framework has to consider aspects of media services that may differ from conventional IT-based SOA such as:

1) Asynchronous communication to properly handle the very large amounts of data associated with AV media in a timely manner.

2) Media Container/Media Descriptor to associate AV metadata with AV essence.

3) Media Infrastructure Service (Resource Manager) for appropriate media handling.

4) Media Bus for AV data exchange, in addition to the conventional ESB for XML message exchange.

5) Media Bus M-SLA (Media-Service Level Agreement), in addition to the conventional SLA in ESB.

6) Security guidelines for secure media handling.

Not all of these aspects are addressed in this version of the specification. It is a goal of the FIMS Task Force to address all aspects in future versions.

Media-centric issues are described in Section 5.5.

Figure 2 illustrates the scope of this version of the FIMS Framework:

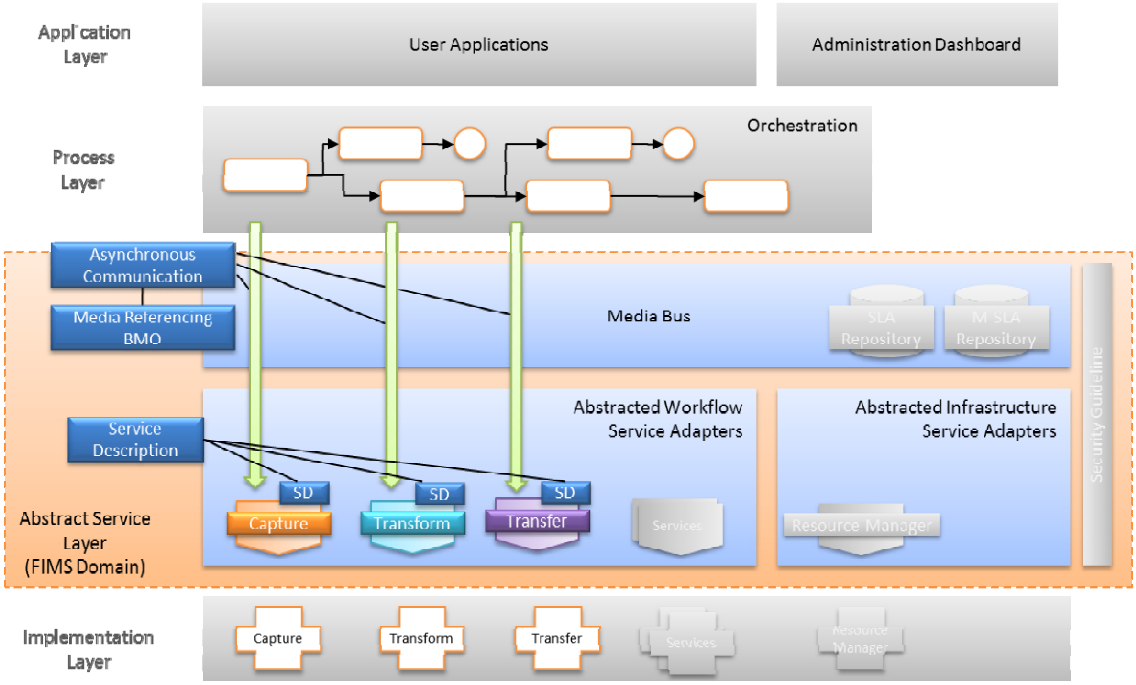


Figure 2: FIMS framework addressed by specification 1.0

The reference architecture is capable of working with different SOA architectural approaches, supporting SOAP/WSDL and RESTful interaction styles.

5.3 Service Taxonomy

A system with FIMS components contains two broad service categories: (1) workflow services able to realize a given business goal (Media Workflow services), (2) infrastructure services that are essential components of the Media SOA system (Media Infrastructure services).

Infrastructure services that are used to construct the Media SOA system are called Media Infrastructure (MI) services. The MI services could (in the future) include a Resource Manager that conducts the resource allocation, as well as other common services like job scheduling and queuing.

Depending on the type of business workflow, the number and type of media services used may vary.

5.4 Scenarios: Mediations, Dynamic resource allocation

The FIMS framework has been conceived in such a way as to allow the implementation of systems with varying levels of complexity, depending on the objectives, operational environment and user expectations.

At the simplest level, services can be invoked directly by consumers (user interface clients or orchestrators), without any intermediate software layer. In this case the workflow orchestrator takes all the responsibility for any supporting operation such as service discovery, resource balancing, etc. In small to medium environments this may be appropriate. In more complex installations, where several processes must run in parallel and compete for resources, it may be convenient to centralize the management operations in an intermediate software layer. This is usually referred to as the Enterprise Service Bus (ESB). Features that can be associated with an ESB include centralised enforcement of security, dynamic resource allocation, service registry management, load balancing, fault tolerance and mediation operations such as protocol translation (for example, from SOAP to REST).

The definition of functionalities that such an intermediate layer should have is outside of the scope of this version of the specification.

5.5 Media Centric Issues

5.5.1 Asynchronous operations for long running process

SOA-based media workflows are often long running processes, sometimes active for hours, days, or even weeks. This places specific persistence requirements of the SOA BPM platform. Servers may be stopped or restarted while processes are running. The system needs to be able to restart at the same point in the workflow and process orchestration without loss or state or data.

Many SOA-based media services run on external hardware or are software-based systems that operate in a loosely coupled asynchronous environment. If the orchestration system is stopped, these services may continue running, and the job process states have to be recovered after the system restarts.

Asynchronous operation status updates may be implemented with either callbacks or polling depending on SOA platform architecture.

The details of asynchronous communication are discussed in Section 8.

5.5.2 Resource Management

5.5.2.1 Process Scheduling and Resource Management

Because of long processing time resulting from the huge size of AV data, process scheduling and resource management can be crucial for SOA-based media systems.

Although the concept of the Resource Manager Service (see Figure 3) includes process scheduling and resource management, process scheduling may be invoked by only the orchestration system. The Resource Management service may be invoked by the orchestration system, by the mediator in the middleware, or by any of the workflow services.

The interface for a Process scheduling service could be hidden if it is only invoked within the closed orchestration system because it is left to the orchestration system developer. However, if it is also to be used by the workflow services, the Resource Manager interface will need to be defined as a part of the FIMS framework. This is not addressed in the current version of the specification.

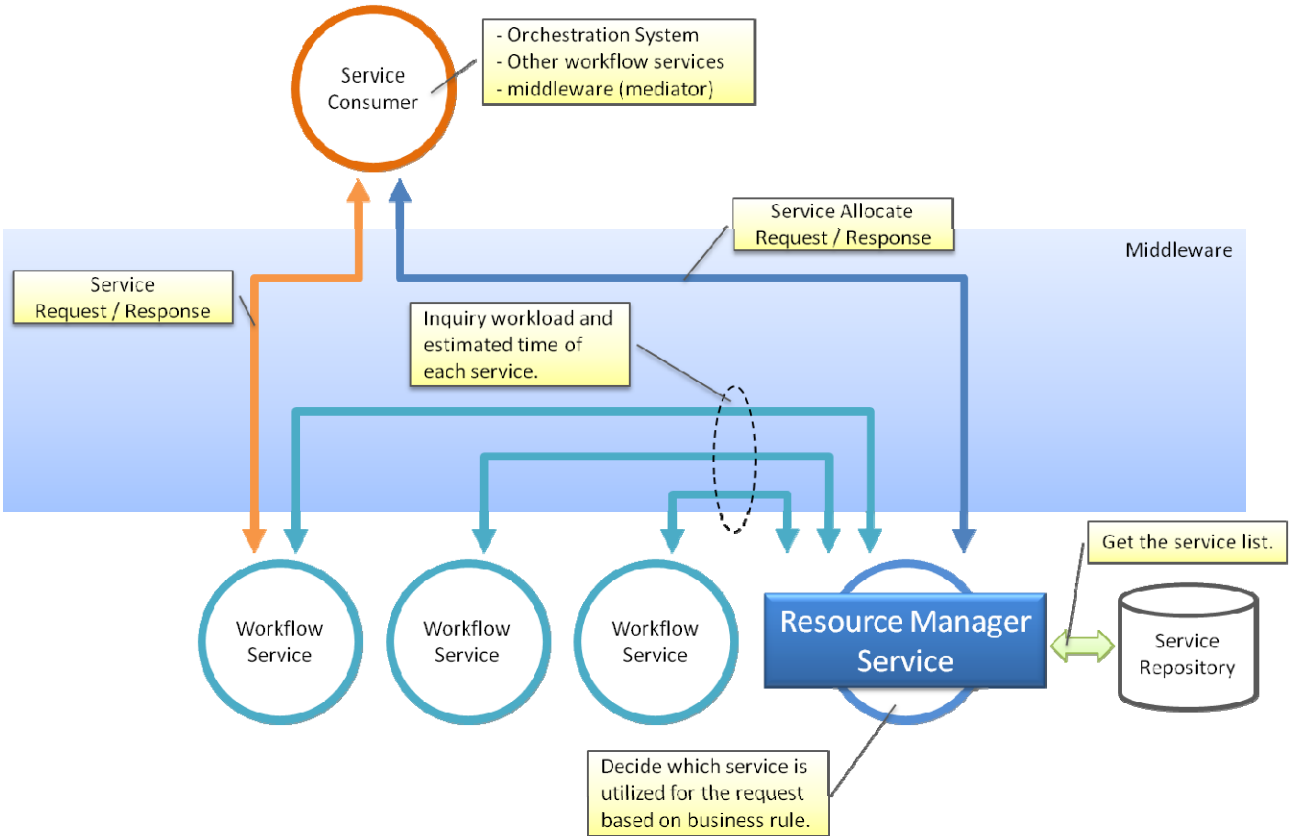


Figure 3: Resource Manager Concept

5.5.3 Media Bus

An extension to SOA often called the "Media Bus" can facilitate storage and media file centric operations. The Media Bus extension is similar to an Enterprise Service Bus (ESB) optimized for large files.

Since the system manages the storage and movement of a very large number of large files (sometimes millions of files per project) it is important that an asset data repository be available as part of the system core functionality, or as an external (third party) service.

There can be many copies of file instances and also many versions located in multiple storage areas. There may also be many lower resolution video proxy files in the system representing the original high resolution "camera negative" files or file sequences. It is necessary to be able to keep track of all these proxy files, including the different versions that may exist.

5.5.3.1 M-SLA

The performance of the individual components of the Media Bus (mainly storage devices) such as bandwidth, transfer speed, capacity, may need to be queried using a common message format.

Based on this, a client such as the orchestration system and/or the media infrastructure service can select the storage device which is best suited to the client, and/or more precisely estimate the execution time to be requested.

This corresponds to SLA in a conventional IT-based SOA system and is referred to as M-SLA. Because the detailed specifications of M-SLA need to be harmonized with SLA, this will need to be need to be discussed by the FIMS Task Force at a later date in accordance with SLA developments.

5.5.4 Security

Due to the high value of the intellectual property passing through media systems, it is critical that security be maintained and access provided only to those with proper authorization. There are several types of security that can be implemented across the SOA: agent-based security, message based security, watermarking, and Digital Rights Management. Typical media enterprises will require most if not all of these security provisions. Agent-based security involves keeping track of the various participants in the SOA-based media system, and doing this correctly will no doubt require some sort of identity management infrastructure.

Identity management technology is well developed in IT, and can be put to very good use in SOA middleware. Instead of using disparate repositories and application-specific methods to authenticate users and secure systems, identity management tools allow the integrator to unify all of an enterprise under a single repository and management system of user data. This allows easy changes to user information and quick provisioning of new users. In an integrated SOA, identity management solutions also allow for role-based views into data.

It is intended that a future version of the FIMS specification will provide guidelines on selection of appropriate technologies from existing security standards.

6. Media Service Management

6.1 Service lifecycle

The main lifecycle states of a service are:

- Service deployed
- Service updated
- Service replaced
- Service decommissioned

6.1.1 Deployment

A service is deployed when the new service or service instance is registered in the service registry. The information provided in the registry includes:

- Service interfaces.
- Service endpoint information.
- Service description metadata.
- Service policies.

6.1.2 Decommissioning

Decommissioning services from the environment is supported by the service registry, which allows the system administrators to first deprecate existing implementations so that potential new consumers do not see the specific implementation. Administrators can then use reporting and impact analysis capabilities in the service registry to allow the operations team to identify remaining service version consumers and ensure that they migrate onto the correct alternative version. After all of the consumers have been migrated, the old service version can be retired from use and removed from the environment.

6.1.3 Replacement/Upgrade

A service can be updated or replaced by updating the service information in the service registry.

The service interfaces and schemas as specified in this FIMS specification shall not require updating if a service is replaced by one providing the same business functionality with the same version of the interface. However, as FIMS specifications evolve, new versions of an interface may be published and may need to be updated in the registry. As service versions are superseded by new implementations which deliver the same required business capability, the service governance lifecycle should allow older versions to be deprecated and, ultimately, retired.

6.1.4 Backward compatibility

Backward compatibility issues might arise for a variety of reasons. This version of the specification focuses on two scenarios:

- There is a new version of the FIMS specifications for the service interface.
- A service implementation has been modified for adding or removing features (such as operations).

6.1.4.1 New Version of the Service Interface

The following practices shall be followed:

- Interface documents and schemas shall have a version with a major and minor number.
- A version attribute shall be used to specify the schema version number.
- A compatible change shall be at least backward compatible. However, if at least a new element is added, it shall be defined so that it allows forward compatibility.
- If a change is compatible, the minor number of the version shall change, and the namespace is not changed. If a change is incompatible, the major number of the version shall change, and the namespace shall change.

If available, the service registry should support the service update lifecycle by allowing the version to be specified for each imported service interface document and schemas.

6.1.4.2 New service implementation

In this scenario there is no change in the version of the interface, but the service is replaced with a different implementation in which either only some of the features (e.g., operations) are implemented or additional new features are implemented. In the latter case, the service is backward compatible. In the former case, there is the issue of making the framework aware that the service is not implementing some features.

The following practices shall be followed:

- Service description for each service implementation provides a list of available service operations and properties (e.g., a transform operation provides a list of supported formats).
- A new service may be invoked through mediation in the ESB or via an orchestration system, in which case a lookup in the service registry can be used to find the service instance(s) that implement the requested operation.
- If no instance with the required operation is found, an error shall be indicated (e.g., service with requested feature not found).

6.2 Job management

6.2.1 Lifecycle of a job

Figure 4 shows the states associated to a long-running job since its request until its completion, cancellation or failure. It also shows the job commands or actions that initiates a transition to a new state.

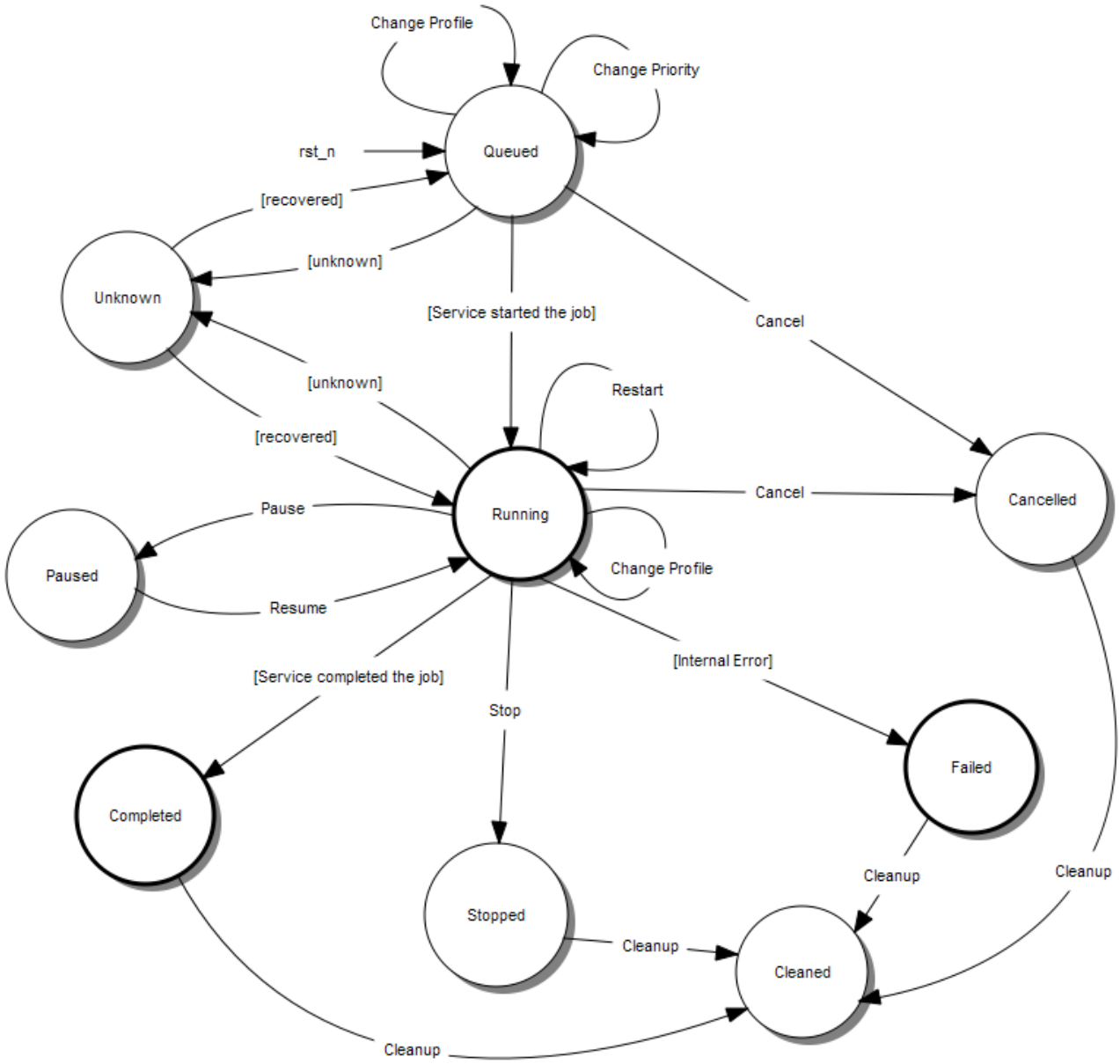


Figure 4: States associated with a long-running job

The thick circles in Figure 4 indicate the mandatory states that shall be implemented.

When a job request is received by a service its initial state is either *Queued* or *Running*. In the *Queued* state a job may have its priority changed before starting its execution through external job management commands (if the service supports these features). The job may also be cancelled by an external command. Once the job is de-queued for execution it moves to *Running* state. In case the service does not implement the job queue the job goes directly to *Running* state upon the

arrival of a request.

The *Running* state indicates a job is being executed. An external command may pause the job (and resume it later). The request may also be restarted or cancelled by external commands. Restarting means to start its execution again. If a job executes until its completion it transitions to *Completed* state. If an error occurs during its execution it moves to *Failed* state. An external command may force the early termination of a job execution. To stop a job means to force its completion. This is not an error situation and the result of the job processing until that moment is considered to be the result of the job execution.

Media services often produce large media files that need to be available until the client or other services retrieve them. Once the job completes (or it is stopped) the resulting product of the operation shall be kept by the service. An external (or internal) command (cleanup) is used to indicate to the service that the result is no longer needed and the job transitions then to *Cleaned* state. Services may independently move jobs to the *Cleaned* state after an elapsed time, the duration of which is to be determined by the service. Services shall be responsible for cleaning jobs across service stop or crash events.

The transition to *Cancelled*, *Completed*, *Stopped* and *Failed* states shall produce a notification to the endpoints specified at the *notifyAt* parameter of the request message, if a notification is expected at the end of the job execution or at the job cancellation. *Completed*, *Stopped* and *Cancelled* shall produce a response notification message to the *replyTo* endpoint, while *Failed* shall produce an error notification message addressed to the *faultTo* endpoint.

The *Unknown* state indicates the job is currently being processed but its state cannot be obtained.

6.2.2 Management Commands

The FIMS services interfaces provide three job management operations. They are the manageJob, manageQueue and queryJob operations, as shown in Figure 5.

MediaServiceStatus		
manageJob		
input	in	manageJobRequest
output	out	manageJobResponse
fault	fault	fault
manageQueue		
input	in	manageQueueRequest
output	out	manageQueueResponse
fault	fault	fault
queryJob		
input	in	queryJobRequest
output	out	queryJobResponse
fault	fault	fault

Figure 5: Job management operations

The *manageJob* operation allows a requester to send job commands to change the state of a job as described in the previous section.

If the service implements a job queue, this queue can be managed using the *manageQueue* operation. The queue commands and the associated state transitions are described in Section 6.2.1.

The *queryJob* operation returns information about jobs that were submitted to the service. The requester may provide a list with the ID of the jobs it wants the information on.

Alternatively, it can request information of jobs that meet requirements specified by a filter object.

The response list contains the identification of the jobs and detailed information about each job such as *estimatedCompletionDuration* and *status*.

6.2.3 Resource-oriented data model

Messages exchanged about jobs, the services they perform, their profiles, queues and the objects that they operate on conform to a resource-oriented data model. The classes of this data model and their relationships are shown in the UML class diagram in Figure 6. The non-referential properties of each class are not shown.

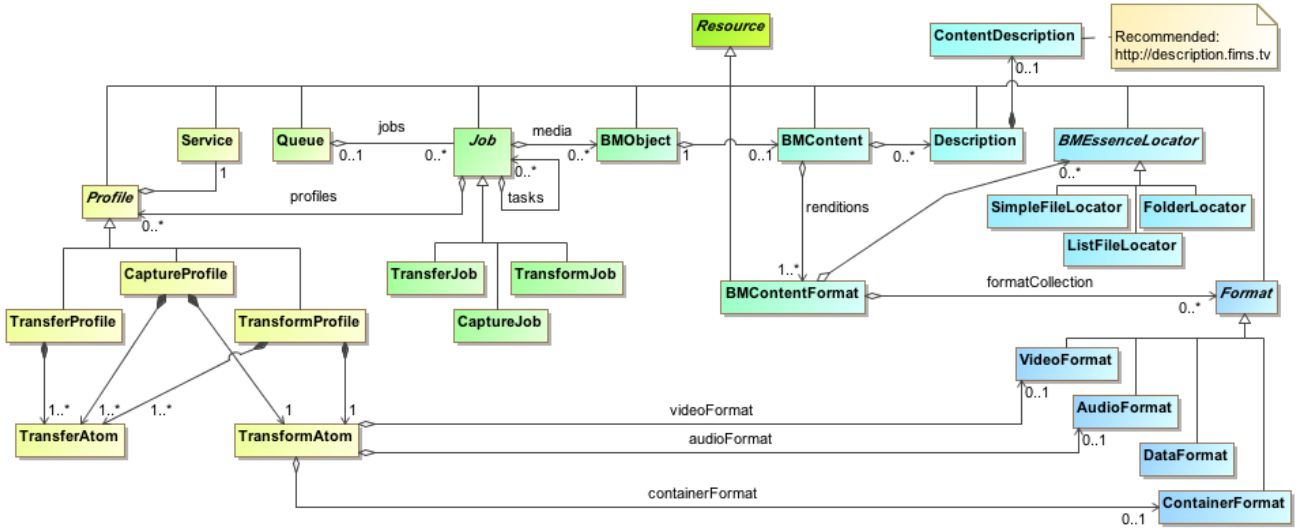


Figure 6: FIMS resources data model

Future extension of the FIMS framework is expected to take place through extension of this data model. For example, BMObject could be extended to provide different kinds of collections of BMObjects that are used as inputs to, or appear as outputs from, services.

Each resource has a unique identifier property “resourceID” that can be used to make reference to it, just like a hyperlink. In combination with a means to resolve references, a resource-oriented approach enables the following:

- reference can be made to shared and repeated resources so that smaller messages can be exchanged. For example, a common profile can be defined for a regularly repeated transform operation and stored in a central location. It does not have to be repeated in every job request, response or status query.
- the location of a resource to be separated from its use, facilitating geographic scalability and greater resilience trough the use of resource-resolution technologies such as DNS.
- support for the introduction of data authority services that provide a single point of authority for classes of metadata. For example, a digital asset management system is used as the data authority for information about content. Rather than a copy of the information about the content being kept by each of the systems involved in a job, with all the associated overhead of keeping a copy, reference can be made to the authoritative version in the asset management system.
- a greater degree of loose-coupling, including more agility to introduce loosely-coupled

monitoring systems and dashboards.

- RESTful bindings for resources. Note that these bindings are enabled by the data model but are not specified in this version of the specification.

An optional “revisionID” property may be used to keep track of revision numbers for a resource, supporting systems that use an eventually consistent approach.

An optional URI “location” property may be used to provide a specific location for the resource (URL) that can be referenced without the need for resource identifier resolution.

In terms of the normative representation of the data model as an XML schema, all properties for resources are marked as optional from an XML perspective (minOccurs=”0”), with the exception of the “resourceID” property. This mechanism allows a resource to be included by reference rather than embedding it.

To encode a reference to a resource rather than embedding it, omit all of its properties other than “resourceID”, “revisionID” (where used) and “location” (where used). The decoder of the message is then expected to either resolve the reference to the resource or report a fault.

To encode a resource by embedding its value, include the value of at least one property other than “resourceID”, “revisionID” and “location”. In general, all known property values for a resource are encoded to minimize the requirement to merge versions of the resource. The decoder is not expected to resolve the resource by reference externally and may update an internal copy of the resource based on the information provided.

The use of XML optionality (minOccurs=”0”) for a property in the XML schema does not imply that the property is optional in a FIMS implementation. Properties that are defined as mandatory through specification, in either a request or a response message and in at least one job state, must be supported by all FIMS implementations.

Note: Mandatory properties are in request or response messages are defined per property in Part 2 of this specification and in the XML schema annotations with source "urn:x-fims:inclusionInRequest" and "urn:x-fims:inclusionInResponse".

An informative and detailed UML data model showing most of the FIMS data model on a single page is provided in the package with the XML schemas.

FIMS defines both normative technical metadata (e.g., Format, VideoFormat, AudioFormat) and recommended descriptive metadata (ContentDescription). Users are encouraged to use the core descriptive metadata provided in the FIMS schema to improve interoperability. Both technical and descriptive metadata are based on EBUCore, an EBU extension of the Dublin Core for media.

7. Media Service Awareness

7.1 Service registry

A registry is useful for tracking deployed services throughout an infrastructure. By making this registry machine-readable and well defined the overall system can make intelligent decisions. The registration mechanism is out of scope of version 1 of the FIMS specification, it may be a complex data exchange mechanism or a human editing a file.

7.1.1 Listing registered services

A FIMS system may provide a registry, in which case it shall contain a line-delimited list of URLs. This list shall be made available via an HTTP GET.

Each URL within the list shall return a response compliant to the service description document defined in section 7.2.

The URL within the registration file may provide a hint as a 'Service' query string indicating the Service Description that will be discovered when the URL is queried.

For example, the registry could contain URLs such as the following, including for example indicating that the query will return details to discover a Capture service:

http://some.dns.entry/
http://some.dns.entry/AnyURL
http://192.168.1.1:8888/SomeURL?Service=Capture
http://192.168.1.1:8888/SomeOtherURL?Service=Transform

A single description discovery URL shall only describe a single service endpoint and a single capability.

For example, the following is invalid:

http://192.168.1.1:1234/YetAnotherURL?Service=Transform,Capture

7.2 Service description

Services should support a mechanism for publishing their capabilities and configuration to systems (and humans). Without understanding what a service will accept or can process, orchestration choices become constrained and out-of-band decisions are required. Whenever information is moved out-of-band, the ability of a system to automatically react is compromised. Either the system must wait for human intervention, or the information must be brought back into the scope of the system via extension.

For this reason, the FIMS specification defines a machine-readable model for providing information about operations a service will accept along with the current configuration of the service. This model references EBU Core Metadata.

The service registry (section 7.1.1) defines URL locations that shall be used to return a service description XML document. When requested, a service shall return a document describing the capabilities and configuration of that service. Such attributes of a service may change over time, and the document returned may vary during the lifetime of a deployed service, for example, due to system faults, licensing restrictions, software upgrades or system administration restrictions to operations.

As an alternative, QueryCapabilityRequest can be used to obtain the location of service description.

A single service instance may implement multiple classes of operation (such as a service that can perform transformation as well as plain transfer services) the service shall not register these on a single endpoint. Such a service shall register multiple endpoints.

8. Media Service Behaviour

This section addresses the service behavior that is common to all categories of services. Behavior specific to each service category is also described.

8.1 Common Service Behaviour

8.1.1 Resource-oriented dialogue

Communication in FIMS consists of a dialogue of messages about the FIMS-defined resources (job, queue, service, profile, BMObject etc.) between a service and its client. An operation of a media service is executed through a dialogue about a job resource between the service provider and the operation requester.

FIMS defines a set of well-known operation implementation patterns that are supported through the SOAP/WSDL definitions provided, as described in section 8.1.2. The corresponding messages embed representations of resources and/or resource references by identifiers as parameters to operations. The WSDL service definitions provide for synchronous and asynchronous requests, responses, faults and notifications.

A RESTful approach to FIMS may use standard HTTP verbs with URI paths to achieve the same dialogue, with the resource description embedded directly as the message body. The specification of specific bindings between the HTTP verbs and the FIMS-defined operation patterns and a RESTful event mechanism for notifications is out of scope for this version of the FIMS specification.

Note that one service provider could simultaneously support both SOAP/WSDL and RESTful interaction styles for the same resources.

Although FIMS 1.0 uses a resource-oriented approach for its objects, this version of FIMS does not specify the required patterns for a fully RESTful approach. These may be considered for a future version, and are likely to include: patterns for resource URIs, use of HTTP operations, headers and status codes, and the use of hypertext links to allow clients to navigate through the resources they need to access.

Figure 7 shows an example sequence diagram of Capture Request/Ack using the SOAP approach.

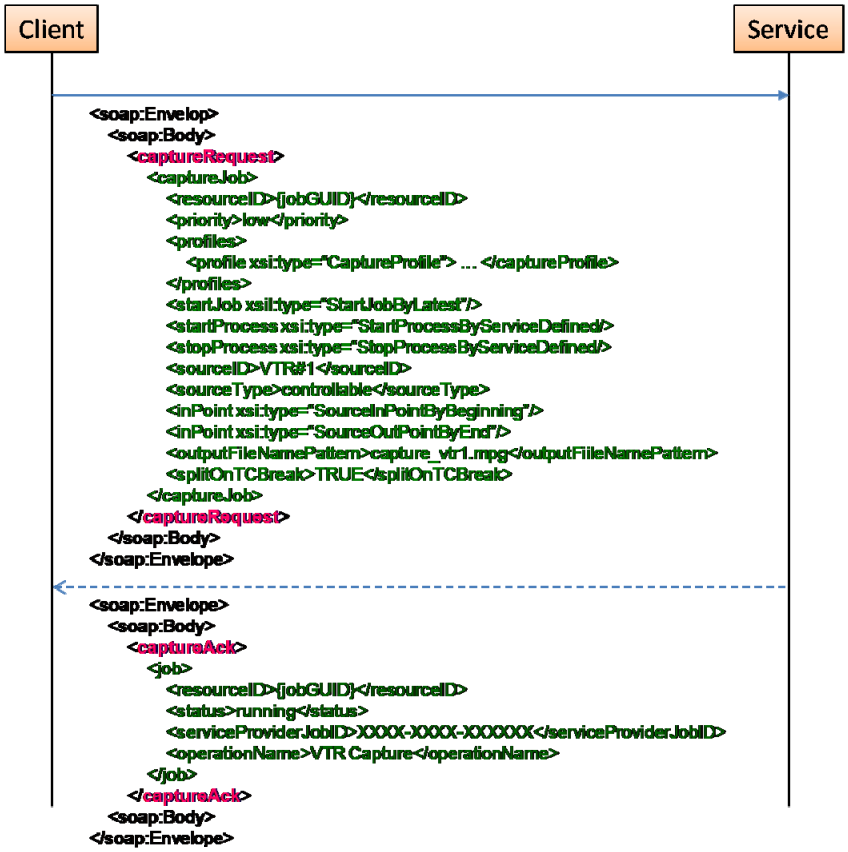


Figure 7: Service Request/Ack (Capture) - SOAP-RPC

8.1.2 Operation Implementation Patterns

Service operations defined by FIMS interfaces provide different types of interactions between the service provider and the service requester. Each one specifies how the result of an operation is made available to the requester. The interface definition along with the input parameters of the operation determines how the service should return the response of the operation.

8.1.2.1 Synchronous Request/Response

In this interaction mode the service client (e.g. a business process) invokes the service to perform an operation passing the input parameters (*par1*, … *parN*) and receives the response in the same communication session as the request. Operations that implement this mode of interaction should not be long-running processes to avoid blocking the service client for a long period of time and to prevent timeouts that may occur in the communication session. See Figure 8.

Figure 8: Synchronous request/response

Examples of operations that use this type of interaction are the job and queue management operations.

8.1.2.2 Asynchronous Request/Response with Notification

This interaction pattern should be used for long running processes. The request and response associated to the operation are exchanged in two different communication sessions. The request session includes the invocation by the client of an operation passing the input parameters (*par1, … parN, jobGUID, notifyAt)* and the acknowledgement by the service that the request was received. A service shall return an acknowledgement when it is ready to respond to any further actions for that request from the orchestration system. See Figure 9.

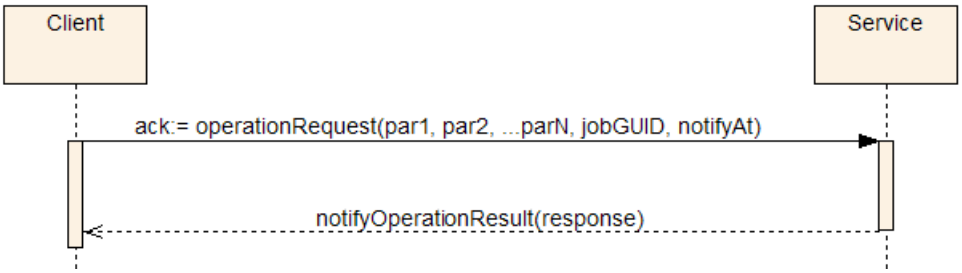


Figure 9: Asynchronous request/response with notification

The *jobGUID* parameter identifies uniquely the job request from the business process (the service client) point of view. The *notifyAt* parameter specifies to the service provider where to send the response message when the operation execution completes. It also specifies where to send an error notification message if the service fails during its execution. The *notifyAt* parameter shall be provided for the service to operate in this mode (see AsyncEndpointType definition).

A separate communication session is used to send the response message to the address specified by the *replyTo* element of the *notifyAt* parameter. If an error occurs during the process of the operation an error notification should be issued to the endpoint specified by the *faultTo* element of the *notifyAt* parameter.

An example of operation that employs this interaction mode is the transform operation of the Transform Media service. See Figure 10.

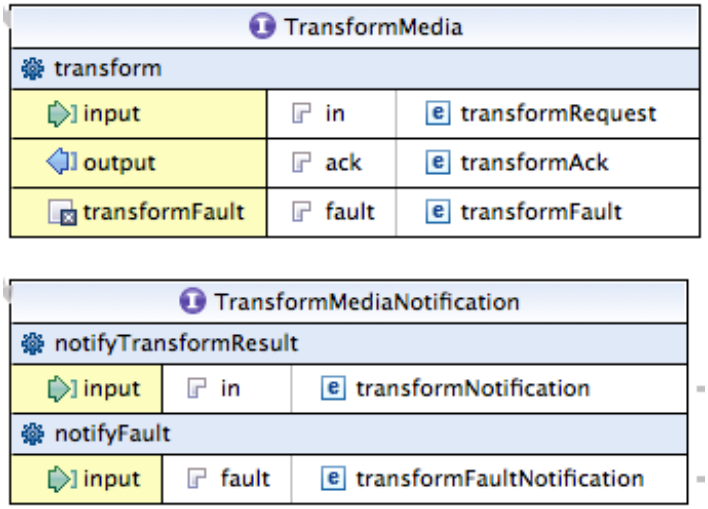


Figure 10: Example of asynchronous request/response with notification: Transform service

8.1.2.3 Asynchronous Request/Response with Polling

This is another interaction pattern for long running processes, where the preferred interaction pattern is not possible (e.g. a firewall preventing a service to call back a client). It is similar to the *Asynchronous Request/Response with Notification* interaction mode with the exception that a notification is not sent by the service when the service completes the operation. The *notifyAt* parameter shall not be provided for the service to operate in this mode. See Figure 11.

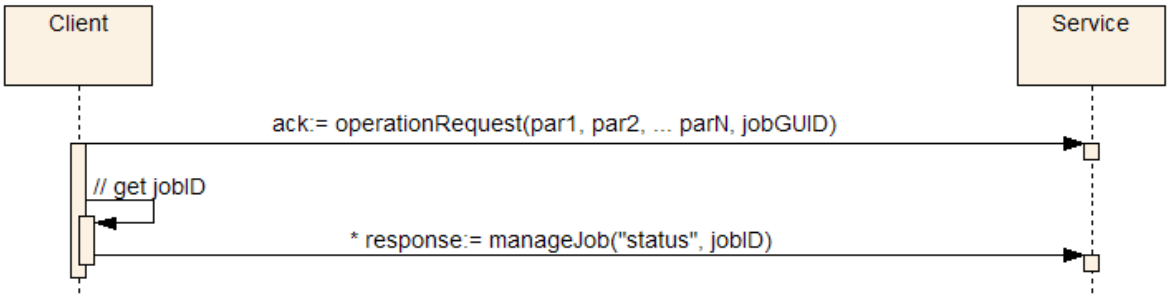


Figure 11: Asynchronous request/response with polling

When the client issues the request, it receives an acknowledgement message that contains the ID of the job as identified by the service.

In this mode of interaction, it is the responsibility of the client to poll the service, using the *queryJob* operation, to retrieve the result of the operation when it is completed. Using the *jobID* parameter (that is part of the acknowledgment message) the client may poll the service to retrieve the status of the requested job. Once the job completes its execution the response message for the status request brings back the result of the operation.

The result of the job execution is contained in the *jobs* element. This element shall be present when retrieving the job information after the service completes the operation. The same information that would be part of the notification message in the pattern specified in Section 8.1.2.2 shall be present in this field.

A transform operation of the Transform Media service may be implemented using this type of interaction.

FIMS services shall support either Notification or Polling.

8.1.3 Input and Output Media

Media services often deal with media files. These services may consume and/or produce files that represent media essences. References to these media files are passed in the input and output messages for these services. Media is represented as by a Business Media Object, as described in Part 2.

8.1.3.1 Processing the Input Media

There are operations that require only a media essence (or list of essence), like transforming media content. In this case the service receives a container object (of type BMObjectType) with a list of content objects that extends the abstract type BMContentFormatType, which represents the media essence(s) the service should operate on.

8.1.3.2 Producing the output content

Services such as Transform Media produce new media essence (or a list of media essence). These services return one or more content objects (of type BMContentFormatType) as a result of the operation. They also shall return a mapping object that represents the relationship between the input content and the output content. This mapping information may be used by external services to keep track of the genealogy and usage of a media essence.

8.1.4 Error and exception handling

Errors and exceptions detected by the service during the execution of job requests shall be returned to the service requester. For synchronous operations as described in Section 8.1.2.1 a fault message shall be returned as a response to the request. See Figure 12.

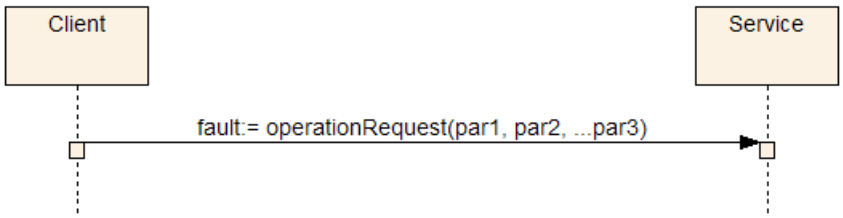


Figure 12: Fault messages handling during job execution

The fault message may contain detailed information about the error. See Figure 13.

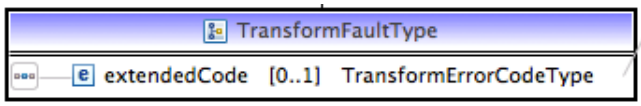


Figure 13: Example of fault message handling

The FIMS framework defines the enumeration of error codes and their description for each one of the service categories. A number of errors are common to all service categories and are presented in **Part 2**.

For asynchronous operation as defined in Section 8.1.2.2 and Section 8.1.2.3, errors may occur in two distinct phases. An error or exception may be thrown at the request time (e.g. Invalid Request Parameters) and a fault message shall be returned immediately as a response to the request. The behavior is similar to the synchronous scenario.

On the other hand if an error is detected during the execution of a long-running process (e.g. job failed) and the notifyAt parameter has been set in the request message, then an error notification message shall be sent to the destination specified by *faultTo* field of *notifyAt*. See Figure 14.

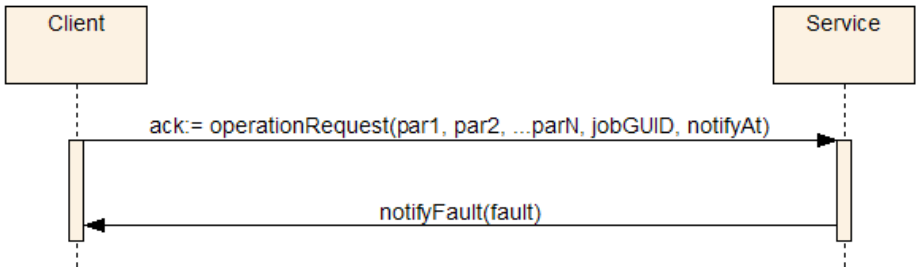


Figure 14: Fault message handling during execution of a long-running process

The recipient of the error notification message shall implement an interface defined by the service. For example, in the case of the *Transform Media* service the recipient implements the *notifyFault* operation specified by *TransformMediaNotification*. See Figure 15.

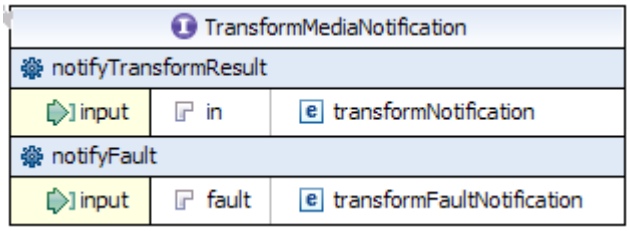


Figure 15: Example of fault message handling with notification

8.1.5 Failure Recovery

A service failure during the execution of a workflow should be a recoverable process. The framework allows for the retry of the failed service since in many cases the cause of the failure can be easily fixed by human intervention. In this scenario the workflow is paused at the invocation of the failed service and can be restarted manually after the service is fixed.

The framework also allows for the definition of a compensation mechanism in the workflow to rollback to a well-defined condition when the failure cannot be recovered by a simple retry mechanism.

8.1.6 Job Queue

A service may implement a queue to support multiple simultaneous requests. If implemented, job requests are en-queued in the order of priority and arrival. The service de-queues the jobs and processes them one by one. Multiple jobs can be de-queued at once if the service supports the execution of multiple simultaneous jobs (multi-threading).

Figure 16 shows the states associated to the job queue and the transitions initiated by the queue commands issued to the service. The states are associated to the processes that en-queue (accepts new jobs) and de-queue (starts execution of an en-queued job) requests. The *Started* state means that both the en-queuing and de-queuing processes are active. *Locked* means jobs cannot be en-queued but they are still being de-queued. *Stopped* means jobs are not being either en-queued or de-queued.

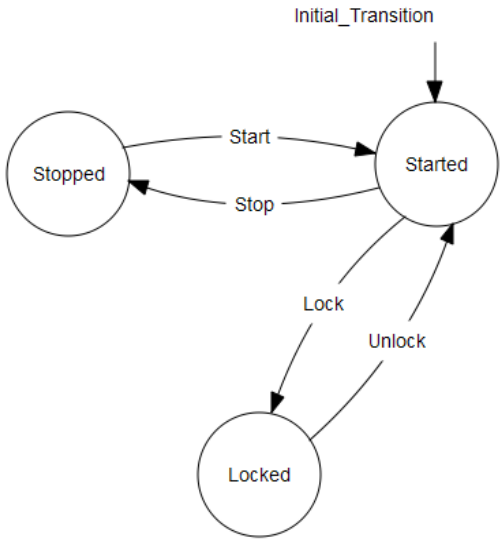


Figure 16: States associated to the job queue and the transitions initiated by the queue commands

The service shall implement the *manageQueue* operation to process queue commands. In case the service does not implement a queue it shall respond with “Feature Not Supported” error code

(SVC_S00_0015).

A job queue may have a maximum size and when it is reached no more jobs will be accepted. A fault indicating the queue is full shall be thrown by the service when a client issues a new job request (SVC_S00_0008).

8.1.7 Job Execution Priority

The service should execute requests in the order of priority. The priority of a job is indicated in the request message. Higher priority jobs take precedence over lower priority jobs.

The framework defines level of priority for the request. The list of possible values for priority are: "low", "medium", "high", "urgent" and "immediate".

- A new job with "low" priority shall be allocated at the end of the queue.
- A new job with "medium" priority shall be allocated to be executed before any "low" priority job but after existing "medium" priority jobs.
- A new job with "high" priority shall be allocated to be executed before any "medium" and "low" priority job but after existing "high" priority jobs.
- A new job with "urgent" priority shall be allocated to be executed before any "high", "medium" and "low" priority job but after existing "urgent" priority jobs.
- A new job with "immediate" priority should be executed as soon as the job request is received.

If the service can process only one job at a time and the job being executed has a lower priority than a new requested job, the existing job shall continue its execution till completion and only then the new job processing should start.

Except for a job with “immediate” priority, prioritization shall only affect jobs that are in the queue. To change what is actually running shall require an explicit operation, which is out of the scope of this specification.

8.1.8 Media Referencing

FIMS media services operate on content essence. In order to describe interoperability of content essence, a unified mechanism to reference essence and metadata is required. This section describes an object model that provides a simple mechanism to support the FIMS requirements: the Business Media Object (BMObject). A BMObject provides business-level information on the media content exchanged by FIMS services (such as location of the media, format and size). The BMObject provides visibility at the business process level of the media content, thus the term 'business' - to indicate that this is a business-level object, which does not overlap with existing standards for the description of media and metadata, but rather provides a convenient mechanism to interoperate transparently with such standards. The BMObject can be as simple as a URL to the media content and a description, yet it is very flexible, since it can represent complex content models and provides a mechanism for transporting metadata as needed.

The BMObject model provides a simple reference to content essence files based on the BMContentType, and a minimum set of descriptive and technical metadata properties, based on EBU Tech 3293 (EBUCore Metadata Set). This can be used by the FIMS framework to make business rules driven decisions (such as selecting the best transform media service for a job based on the content format or the size of the content).

Future extensions of the framework may extend the BMObjectType with different kinds of collections of content. For example:

- Different editorial cuts of the same television program.
- Cuts only sequence input for a transcoder.
- Edit decision lists.
- A complex mapping of the relationship between input and output content.

8.1.8.1 Partial Media Reference

- FIMS specification 1.0 for transfer media and transform media does not require partial media reference
- Capture provides InPoint/OutPoint, which allows specifying partial media capture with time codes.
- Additional services might require the ability to reference partial media.

8.1.9 Jobs

An essential aspect for the management of long-running media operations is the ability to check the status and interact with requested jobs. For example, the following tasks might be performed for transcoding operations:

- Check the status of a job or list of jobs
- Cancel an active job
- Pause an active job
- Stop an active job
- Change priority of a job
- Clear a queue
- Lock/unlock a queue

The adoption of a common interface for the status operations enables the use of common front-end tools for media services management, and the ability to interact with running tasks from a client or workflow. Queue and Job management types provide a mechanism to query and manage the status of jobs and job queues.

8.1.10 Errors

The base schema defines a fault type that can be extended by specific FIMS service classes to provide service-class specific error codes. One of the benefits in utilizing a media service abstract class is the definition of a common set of errors. This definition allows the client requesters to implement different error handling logic for each individual service provider used. With this approach, the client can implement a general compensation or error handling logic for all service providers that support a FIMS service specification.

8.2 Service Interface Overview

This section gives two common functions of service interfaces: Time Constraints and Profiles.

8.2.1 Time Constraints

8.2.1.1 Concept

Time Constraints are time-related constraints during the job execution that are specified in the service request message. Time Constraints consist of the following four parameters:

1) startJob

startJob is one of the elements of the JobType, and is applicable to any service. It specifies the system time when the job shall start. Usually, this indicates the time at which a request message is extracted from the queue and is moved to the running state.

There are the following three types to specify the time parameter of startJob:

- NoWait type: execute immediately
- Time type: the time to start
- Latest type: the latest time at which a process can be started at the startProcess properly.

Supporting types of Time and/or Latest implies that the service supports a special queue which has ability to schedule the job start time according to the value of startJob. A service which supports Time or Latest type shall declare this in the Service Description.

2) startProcess

startProcess is a parameter for services that need to handle a real-time stream function such as Capture or Playout, and specifies the system time at the start of the stream process.

There are the following four types to specify the time parameter of startProcess, which may need to be frame accurate:

- NoWait type: execute immediately
- Time type: the time to start
- TimeMark type: the time at which the TimeMark embedded in essence such as a timecode is detected
- ServiceDefinedTime type: the time defined by a service

A service which supports Time and/or TimeMark type shall declare this in the Service Description.

3) stopProcess

stopProcess is a parameter for services that need to handle a real-time stream function such as Capture or Playout. This specifies the time when the stream process shall stop.

There are the following five types to specify the time parameter of stopProcess, which may need to frame accurate:

- OpenEnd type: the time at which a stop command is received

- Time type: the time to stop
- TimeMark type: the time at which the TimeMark embedded in essence such as a timecode is detected
- Duration type: the time at which point the specified duration has elapsed since startProcess
- ServiceDefinedTime type: the time defined by a service

A service which supports Time, TimeMark, and/or Duration type shall declare this in the Service Description.

4) finishBefore

finishBefore is one of the elements of the JobType, and is applicable to any service. It specifies the time by which the job shall have been completed.

finishBefore also specifies the deadline for the job execution. For example, in the case of specifying TimeMark type in the stopProcess, it can be used as a timeout time. When finishBefore is exceeded before the job is completed, the service shall notify "SVC_S00_0016: Deadline passed" to the orchestration system.

If the service cannot accommodate the time constraint, the service shall notify the error code SVC_S00_0017 to the orchestration system: “Time Constraints in request cannot be met”.

A service that can control the job completion time by using finishBefore shall declare this in the Service Description. Even if a service does not itself support the function to control the job completion time, an orchestration system should specify finishBefore as a deadline.

8.2.1.2 Use Cases on Time Constraints

Table 1 shows possible use cases with the combination of "startProcess" and "stopProcess".

Table 1: Time constraints using a combination of "startProcess" and "stopProcess"

startProcess	stopProcess	Description
NoWait	OpenEnd	Start ASAP, stop when stop command (manageJobRequest) is received.
	TimeMark	Start ASAP, stop when specified timeMark is detected.
	Time	Start ASAP, stop at the specified time.
	ServiceDefined Time	Start ASAP, stop at the service defined time.
	Duration	Start ASAP, stop when specified duration has passed.
TimeMark	OpenEnd	Start when the specified timeMark is detected, stop when stop command (manageJobRequest) is received.
	TimeMark	Start when the specified timeMark is detected, stop when specified timeMark is detected.
	Time	Start when the specified timeMark is detected, stop at the specified time.
	ServiceDefined Time	Start when the specified timeMark is detected, stop at the service defined time.
	Duration	Start when the specified timeMark is detected, stop when specified duration has passed.
Time	OpenEnd	Start from the specified time, stop when stop command (manageJobRequest) is received.
	TimeMark	Start from the specified time, stop when specified timeMark is detected.

	Time	Start from the specified time, stop at the specified time.
	ServiceDefined Time	Start from the specified time, stop at the service defined time
	Duration	Start from the specified time, stop when specified duration has passed.
ServiceDefined Time	OpenEnd	Start at the service defined time, stop when stop command (manageJobRequest) is received.
	TimeMark	Start at the service defined time, stop when specified timeMark is detected.
	Time	Start at the service defined time, stop at the specified time.
	ServiceDefined Time	Start at the service defined time, stop at the service defined time.
	Duration	Start at the service defined time, stop when specified duration has passed.

8.2.1.3 Sequence Diagram Examples

Figure 17 shows an example of the Capture Service sequence diagram in terms of Time Constraints. The source type of this example is VTR and both inPoint and outPoint are the specified time code.

A service receives a request message from an Orchestration System. The service starts job at startJob, and performs some actions such as cueing up to the inPoint, starting playback and detecting specified time code. Important point is that the instruction of how and when to control the device is not included in the request message. The service manages to control the device by itself according to the information such as startProcess/stopProcess, inPoint/outPoint in the request message. A capture process starts at the serviceDefinedTime; it will stop at another serviceDefinedTime. After some finishing job process is performed, a CaptureNortification is issued to the Orchestration System to report completion.

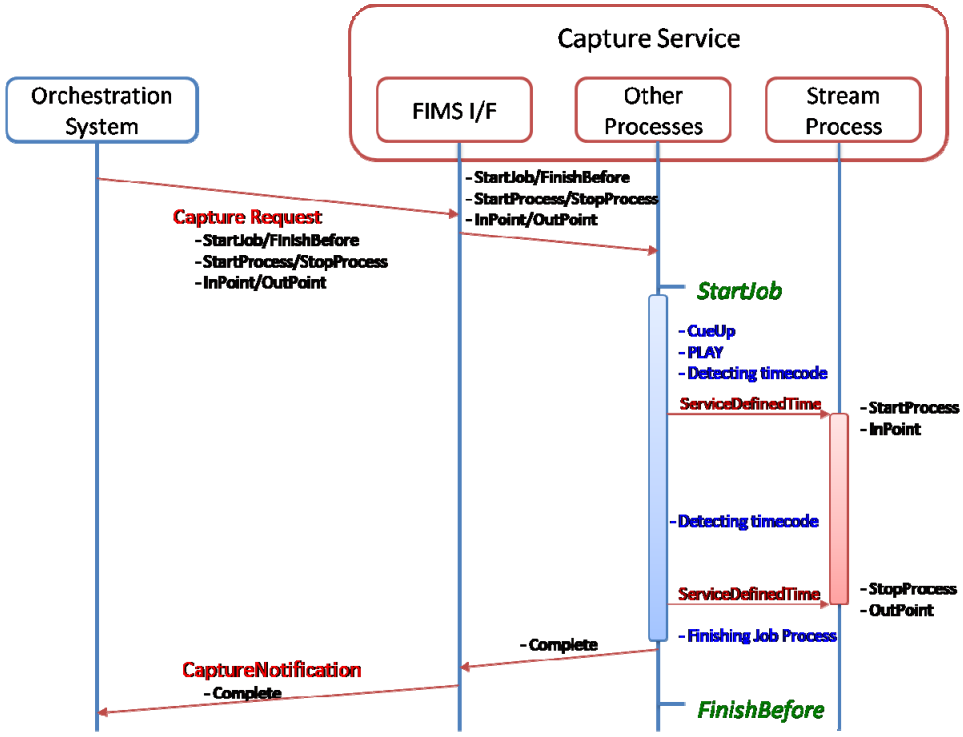


Figure 17: Example of the Capture service sequence diagram in terms of time constraints

Figure 18 shows the case where Time Constraints in the request message cannot be met for some reason. This can occur when a service receives the request message, or during queuing, or even during running. In this case, the service shall issue CaptureFault with the error code

"SVC_S00_0017: Time Constraints in request cannot be met" to the Orchestration System.

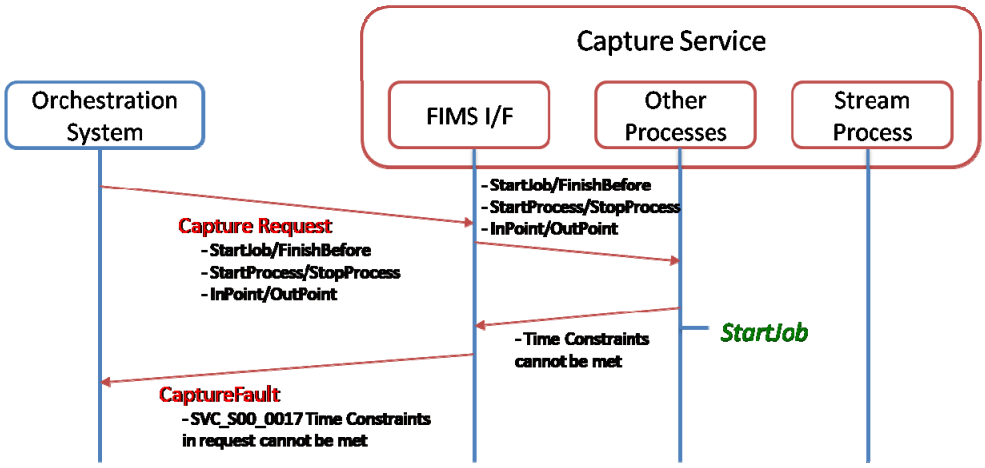


Figure 18: Example where Time Constraints cannot be met

Figure 19 shows the case where the time specified by finishBefore has passed before the capture process has been completed. In this case, the service shall issue CaptureFaultNotification with the error code "SVC_S00_0016 Deadline passed" to the orchestration system. After issuing the CaptureFaultNotification, the service should wait for the next instruction from the orchestration without stopping the process.

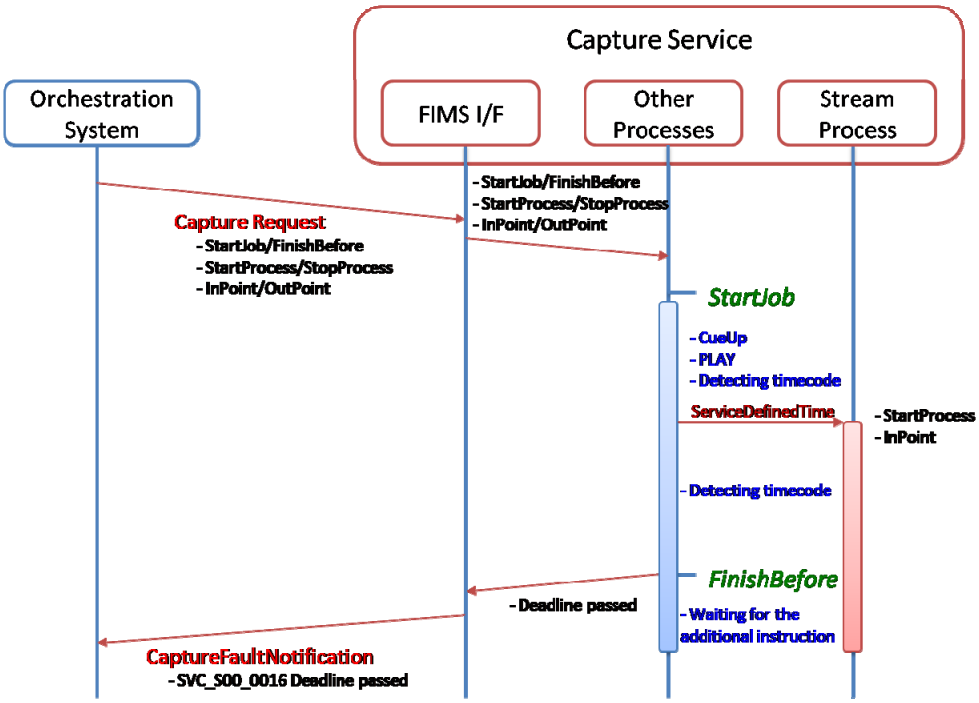


Figure 19: Example where the time specified by finishBefore has passed before completion

8.2.2 Profiles

Requests to any FIMS service interface shall include a profile that describes the specific operations that will take place as part of that request. Profiles themselves are composed of parameters specific to that service interface and generic elements that are available for re-use between interfaces. These reusable elements are combined into groups referred to as Atoms.

Profile structures are specified for each service interface that FIMS defines. Profiles inherit from *ProfileType*, which is used to provide a common basis for all profiles.

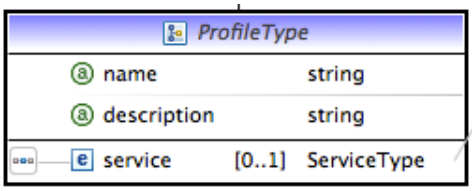


Figure 20: ProfileType

Profiles can be shared between jobs and included by reference. See Section 6.2.3.

Where possible, implementations should use the specified profile.

Figure 21 shows an example of a request with two Capture profiles being used to create main stream essence (J2K + MXF) and proxy essence (AVC + MP4) and outputting though different transfer atoms (possibly to different locations over different transports).

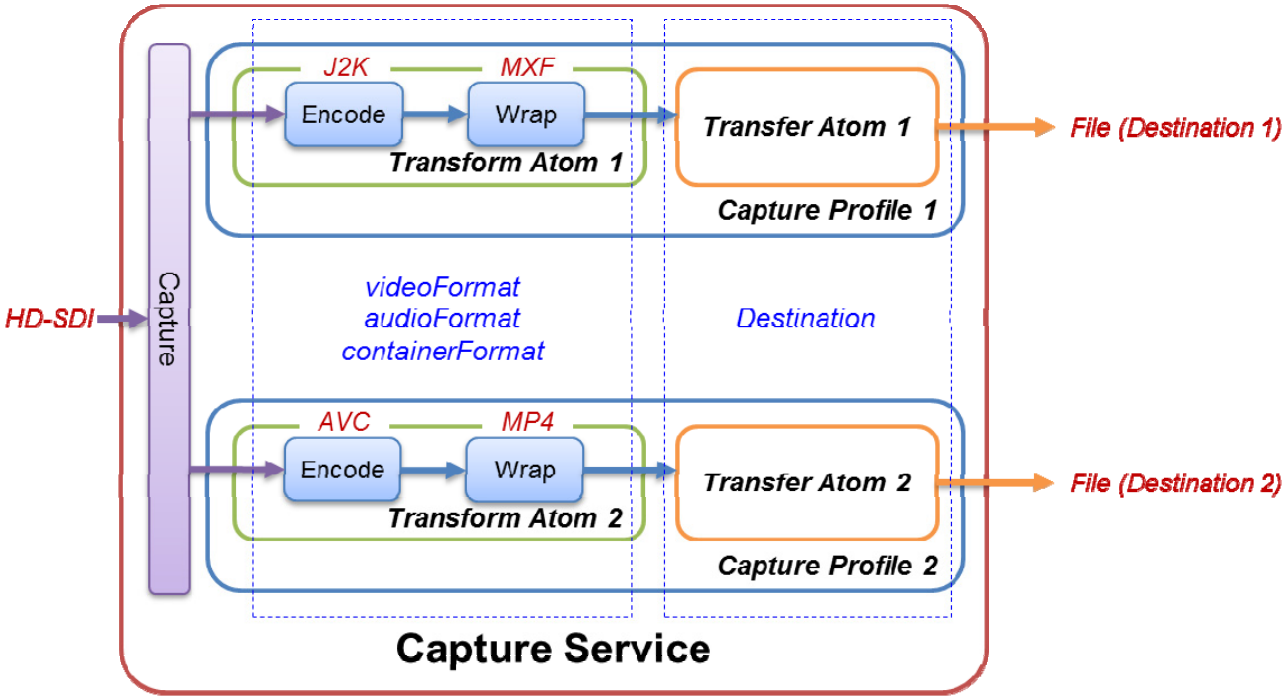


Figure 21: Example of Capture request with multiple profiles

Appendix 1: Future Visions (Informative)

A1. Pipelined Media Services

A Pipelined Media Services is composed of more than one existing service in order to realize functions and/or performance that use of individual existing services by themselves cannot achieve.

As described in Section 8.2.2, a pipelined media service within a service can be realized using profiles. Figure 22 shows an example of extended profile where two Capture (Transform) profiles are used to create main stream essence (J2K + MXF) and proxy essence (AVC + MP4) with AV Process.

AV Process is added in the transformAtom so that processing A/V essence is enabled during capture.

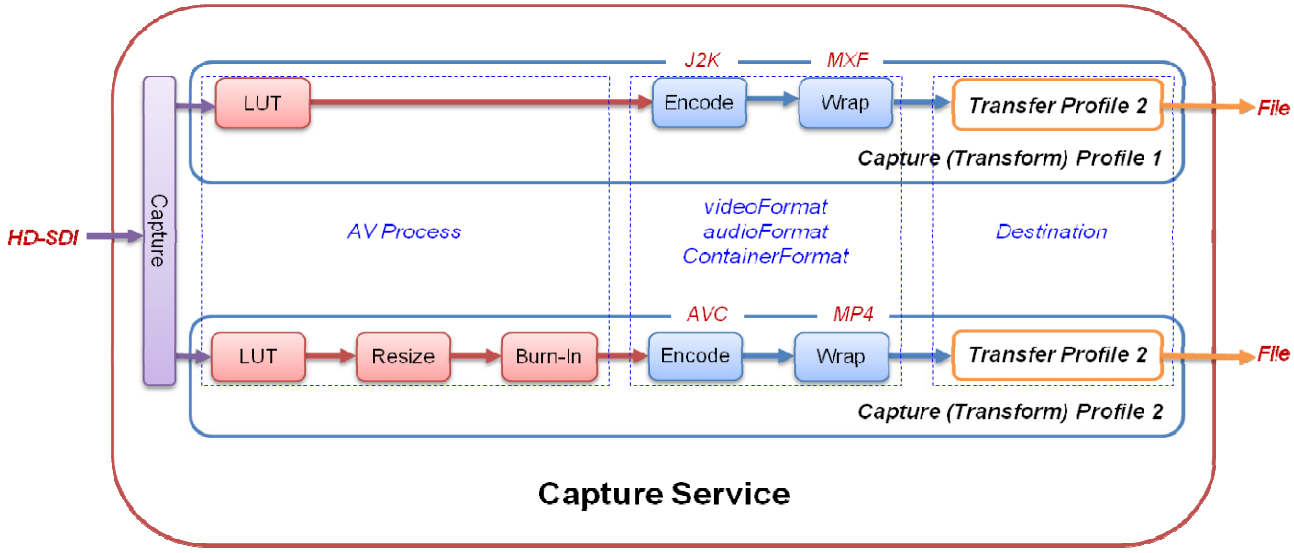


Figure 22: Example of pipelined media service

EBU

OPERATING EUROVISION

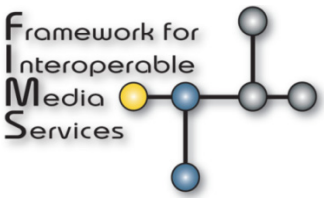
TECH 3356

SPECIFICATION OF THE FIMS MEDIA SOA FRAMEWORK

Part 2: Service Interfaces
S0, Base Schema

VERSION 1.0.7

Published Jointly With FIMS and AMWA



Geneva
September 2012

[*] Page intentionally left blank. This document is paginated for two sided printing

Executive Summary

This document describes a vendor-neutral common framework for implementing Interoperable Media Services using a Service Oriented Architecture (SOA) based system, supporting interoperability, interchangeability and reusability of media specific services.

The FIMS 1.0 release at the time of publication of this document comprises the following:

- FIMS 1.0 Part 1: General Description:
  Part 01-General Description-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S0: Base Schema (this document):
  Part 02_S0-Base Schema-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S1: Transfer Service:
  Part 02_S1-Transfer Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S2: Transform Service
  Part 02_S2-Transform Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S3: Capture Service:
  Part 02_S3-Capture Service-FIMS Schema Spec-1.0.7.pdf
- Schema file package:
  FIMS_1_0_7.zip

Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for two sided printing

Framework for Interoperable Media Services
FIMS Media SOA Framework 1.0

Part 2, Supplement 0: Base Schema

S0 1. Schema Design

The FIMS object model is described by a set of XML schemas, which provide the object model representation for common objects and extensions for the different classes of service. Describing the object model as an XML schema has the advantage of providing a binding-independent representation. The following sections do not describe a specific binding (such as SOAP/HTTP or REST) but rather focus on the types used to construct the job request and response messages. This approach has the advantage of describing the structure and the semantic of messages without placing specific binding constraints.

S0 1.1 Base Schema

FIMS services share a common object model, which serves two main purposes:

- Promote interoperability of data objects between different classes of services;
- Promote code reuse. Code libraries that manipulate shared data objects can be reused by different classes of services.

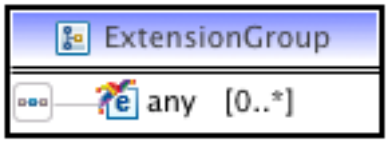
Each FIMS service extends these common object types defined in the base schema. These base types model job requests and responses for long running, asynchronous operations and provide support for job and queue management tasks.

This base schema will be provided as a xml schema file separately and defines types described in the following sections.

S0 2. Base Service Interface

S0 2.1 Extension types

S0 2.1.1 ExtensionGroup

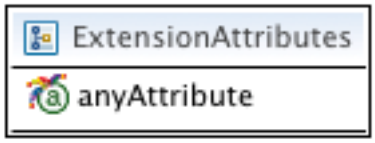
complex type	
Description	
Each Type includes an ExtensionGroup to allow for vendor-specific extensions, but the definition is out of scope of the FIMS specification.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	
	

any

element (type)			
Description			
Any number of extension elements from an external namespace. The lax processing model indicates that if the schema for the extension elements is known, then they should be validated. If the schema is not known, no error will be generated. Extension nodes must be the last element in a sequence of a non-abstract type.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	any	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.1.2 ExtensionAttributes

complex type	
Description	
Each Type includes ExtensionAttributes to allow for vendor-specific extensions, but the definition is out of scope of the FIMS specification.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



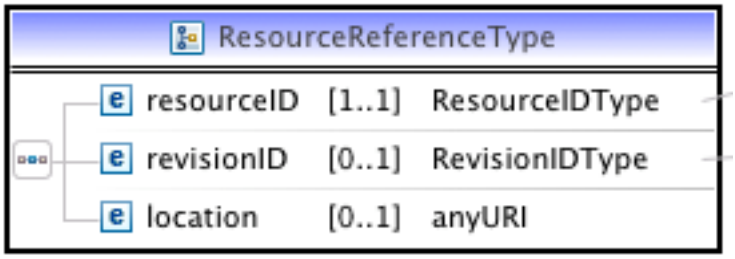
anyAttribute

element (type)			
Description			
Permitted extensions to elements that permit extensions. Attributes from other namespaces are permitted. Lax processing requests that a validator checks the attributes where access to the attribute's defining schema is available.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	anyAttribute	Optional	Optionsl
Service Description	**Contents of Service Description**		
Normative Requirements			
Extension attributes should not be used where an equivalent property is available in the FIMS data model. Attributes from the FIMS base namespace shall not be used.			

S0 2.2 FIMS-defined references

S0 2.2.1 ResourceReferenceType

complex type	
Description	
The source or target of a reference to a distinct resource. Resources include queues, jobs, assets, descriptions, formats and profiles.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



resourceID

element (type)			
Description			
Locally distinct identifier for the resource that can be used to make reference to the resource. The reference may also be globally unique.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:ResourceIDType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

revisionID

element (type)			
Description			
Identifier for the specific revision of the resource. Tracking the different versions of resources across systems enables different instances and versions of a resource to coexist at different locations and updated to be eventually consistent according to local policy.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:RevisionIDType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
A reference to a resource that omits the revision identification of the resource shall indicate that the latest version is required.			

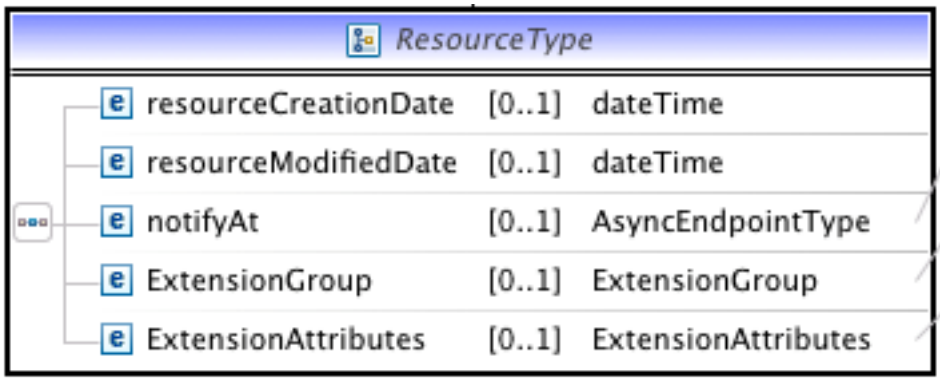
location

element (type)			
Description			
A Uniform Resource Location that points to a specific instance of a resource at a location.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
Reference to a resource that omits the location shall require resolution of the resourceID according to local policy.			

S0 2.3 FIMS-defined resources

S0 2.3.1 ResourceType

complex type (abstract, base)		
Description		
Resources include queues, jobs, assets, descriptions, formats and profiles.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceReferenceType		
Normative Requirements		
Class Diagram		



resourceCreationDate

element (type)			
Description			
Date and time when this resource was created.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

resourceModifiedDate

element (type)			
Description			
Date and time when this resource was last modified. Modification date of the resource should be updated when any of the values of the resource's properties are updated.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

notifyAt

element (type)			
Description			
Endpoints where a service can send back a notification for a job completed or failed.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:AsyncEndpointType	Optional	Not applicable
Service Description	**Contents of Service Description**		
Required	Support or not, including the maximum number of notification endpoints.		
Normative Requirements			
A service shall support either notification or polling. If this element is set, the service shall send a notification to the address(es) indicated in the "replyTo" field(s) for successful completion and shall send a notification to the address(es) indicated in the "faultTo" field(s) for failure. If this element is not set, the service shall not send notifications to the requester			

ExtensionGroup

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup		
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

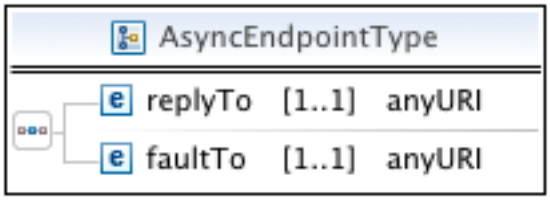
Element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes		
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.2 AsyncEndpointType

complex type
Description
Provides endpoints where a service can send back a notification for a job completed or failed.

Service Description	Content of Service Description

Normative Requirements

Class Diagram


replyTo

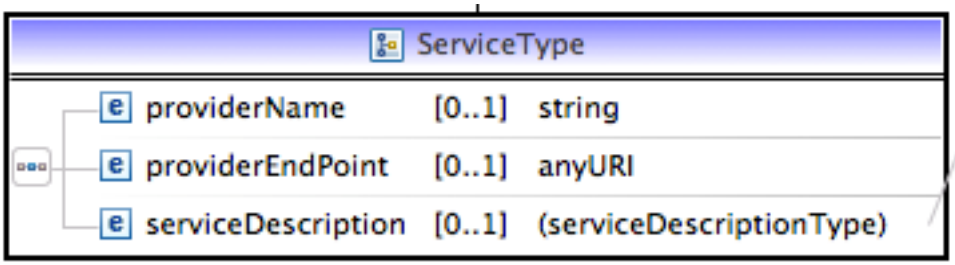
element (type)			
Description			
Endpoint where the service sends notifications indicating the successful termination or cancellation of a job. Depending on the invoking client or business process, the endpoint for "faultTo" can be the same as the "replyTo" or it can be different.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	anyURI	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

faultTo

element (type)			
Description			
Endpoint where the service sends the fault notification indicating the failure of a job. Depending on the invoking client or business process, the endpoint for "faultTo" can be the same as the "replyTo" or it can be different.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	anyURI	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.3 ServiceType

complex type (base)		
Description		
Describes the location endpoint, provider and Service Description of the service.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Class Diagram		



providerName

element (type)			
Description			
Name for the provider of a service.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			

providerEndPoint

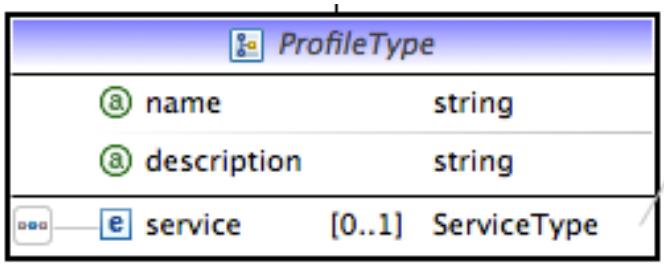
element (type)			
Description			
Provider endpoint that uniquely locates the provider servicing the job request.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	anyURI	Optional	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

serviceDescription

element (child)			
Description			
Service-specific description of the capabilities offered by the service. The transfer, transcode and capture services defined alongside the original FIMS framework have service descriptions provided and, if used, it is recommended that these are embedded here.			
Occurrence	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	any, 1..*	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.4 ProfileType

complex type (abstract, base)		
Description		
A profile used by a FIMS media service to perform a job on media content. For example, it may represent the profile of a transfer media service for transferring media content and, as such, may specify the media format to be produced in output. The profile provides a mechanism to specify service-provider-specific information for each operation.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Class Diagram		



service

element (type)			
Description			
Details of the service that has been selected to execute this profile.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ServiceType	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

name

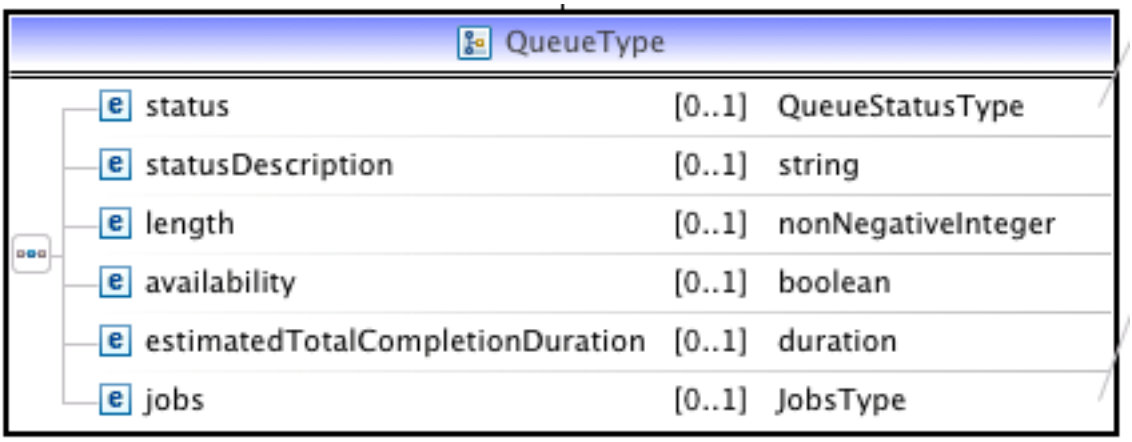
attribute			
Description			
Name of the profile.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

description

attribute			
Description			
Description of the profile.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.5 QueueType

complex type (base)		
Description		
Provides basic queue information, such as the status and length of the queue.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType	Required	Support or not the queue, queue depth, and supported queue commands
Normative Requirements		
Class Diagram		



status

element (type)			
Description			
Status information for the queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:QueueStatusType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

statusDescription

element (type)			
Description			
An optional description for the status.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

length

element (type)			
Description			
Length of the queue, measured in the number of jobs currently in this queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

availability

element (type)			
Description			
Determines if the queue is currently available.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	boolean	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

estimatedTotalCompletionDuration

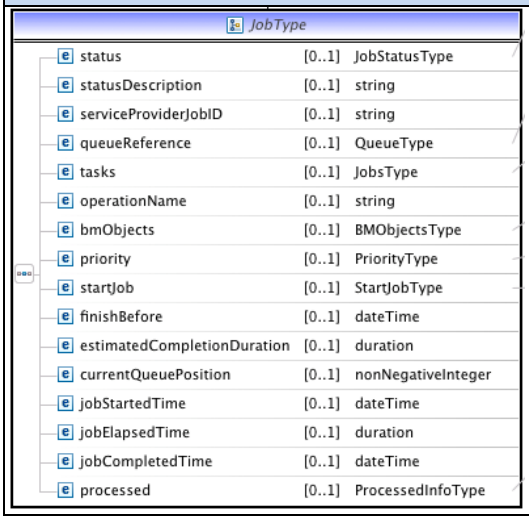
element (type)			
Description			
Estimate of the time duration required for processing all jobs currently in the queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	duration	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			

jobs

element (type)			
Description			
Optional list of jobs currently in the queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:JobsType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.6 JobType

complex type (abstract, base)		
Description		
Describes a job, which is extended in each service schema.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType	Required	Supported job commands
Normative Requirements		
Class Diagram		



status

element (type)			
Description			
The current status of the job (e.g. running, completed), enumerated in JobStatusType.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:JobStatusType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

statusDescription

element (type)			
Description			
Optional description for job status.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

serviceProviderJobID

element (type)			
Description			
Service-provided local job identifier, such as a job identifier provided locally by an application.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
If the service provider generates its own local job identifier associated with a job request, this field may be set. Otherwise, the service provider may use the supplied resource identifier.			

queueReference

element (type)			
Description			
The queue that has master control over execution of this job. As a FIMS-compliant device is not required to support queues, this element is optional.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:QueueType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
The property shall be used as a reference to a queue containing the resource ID of the queue.			

tasks

element (type)			
Description			
A list of sub-tasks that consists of related jobs running in parallel. For information only in FIMS 1.0. Whether the sub-tasks are running in parallel or sequence to one another is not recorded in this model.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:JobsType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

operationName

element (type)			
Description			
Descriptive name of the operation associated with this job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

bmObjects

element (type)			
Description			
Business media objects produced or consumed by the job. For some services, such as a capture service, no BMObjects are consumed. However, the job will include a BMObject on completion.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:BMObjectsType	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

priority

element (type)			
Description			
The priority for the job (e.g. "immediate", "high"...) enumerated in PriorityType. Where supported, job priorities describe the relative priority of a job in the owning queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:PriorityType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			
The priority shall be included where the job is being managed as part of a queue.			

startJob

element (type)			
Description			
Parameter used to indicate that the job shall be initiated no earlier then the specified time.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:StartJobType	Mandatory	Optional
Service Description	**Contents of Service Description**		
Required	Support or not and supported StartJob types		
Normative Requirements			
A service shall support at least the NoWait type.			

finishBefore

element (type)			
Description			
Specifies the time by which the job shall have been completed. This element also specifies the deadline for job execution. For example, in the case of specifying "TimeMark" type in the "stopProcess", it can be used as a timeout time.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			
When "finishBefore" is exceeded before the job is completed, the service shall notify "deadline passed" to the orchestration system. If the service cannot accommodate the time constraint, the service shall notify "time constraints in request cannot be met" to the orchestration system. Even if a service does not itself support the function to control the job completion time, an orchestration system should specify "finishBefore" as a deadline.			

estimatedCompletionDuration

element (type)			
Description			
Before the job has started an estimate for the expected total duration of the job. Once the job has started, it provides an estimation of time remaining until the end of the job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	duration	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			

currentQueuePosition

element (type)			
Description			
The position of the job in the queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			
The current queue position shall be included where the job is being managed as part of a queue.			

jobStartedTime

element (type)			
Description			
The date and time this job started.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

jobElapsedTime

element (type)			
Description			
The time elapsed since the job started.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	duration	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

jobCompletedTime

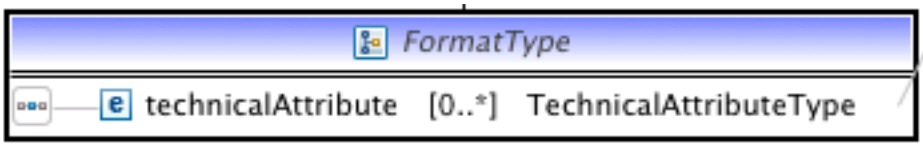
element (type)			
Description			
The time and date that the job completed.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			

processed

element (type)			
Description			
Provides statistics on number of bytes or frames processed. Note that a time-based percentage value to completion is not directly provided, but can be calculated if the service provides the "estimatedCompletionDuration". The time-based percentage is a ratio between "jobElapsedTime" and "estimatedCompletionDuration".			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ProcessedInfoType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not and which kind of processed information is provided.		
Normative Requirements			

S0 2.3.7 FormatType

complex type (abstract, base)		
Description		
Characteristics of files, video, audio and ancillary data.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Class Diagram		

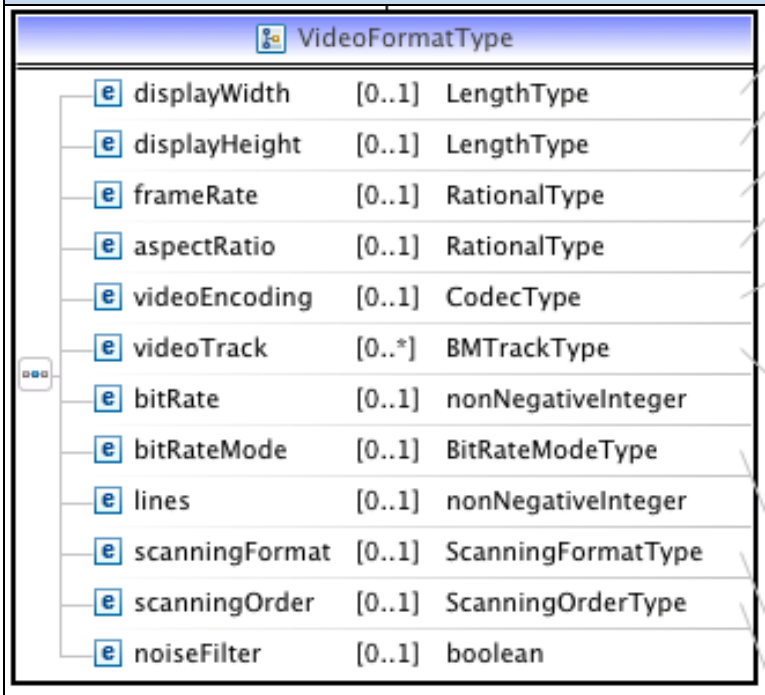


technicalAttribute

element (type)			
Description			
User defined technical attributes. See 'technical attribute' in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	bms:TechnicalAttributeType	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.8 VideoFormatType

complex type (base)		
Description		
Characteristics of a video signal. See 'videoFormat' in EBU Tech 3293 for more information.		
Base	**Service Description**	**Content of Service Description**
bms:FormatType		
Normative Requirements		
Class Diagram		



displayWidth

element (type)			
Description			
Width of the image or picture. Used as numerator to define the aspect ratio for video content. See 'width' in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:LengthType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

displayHeight

element (type)			
Description			
Height of the image or picture. Used as denominator to define the aspect ratio for video content. See 'height' in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:LengthType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

frameRate

element (type)			
Description			
Frame rate for the video content, measured in frames/second.			
Occurrence	Type	Inclusion (Req.)	Inclusion (Res.)
0..1	bms:RationalType	Optional	Optional
Service Description	Contents of Service Description		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

aspectRatio

element (type)			
Description			
Ratio of the picture (the width by the height), for instance '4:3' or '16:9' (rational). The format of the aspect ratio is specified in the format attributes. See 'aspectRatio' in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:RationalType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

videoEncoding

element (type)			
Description			
Defines the encoding parameters of the resource (e.g. H264) for a video channel. See 'videoEncoding' in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:CodecType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

videoTrack

element (type)			
Description			
Definition of the video tracks in the source.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	BMTrackType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

bitRate

element (type)			
Description			
Number of bits at which the video is encoded, measured in bits per second.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

bitRateMode

element (type)			
Description			
Type of bitrate, either constant or variable.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:BitRateModeType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

lines

element (type)			
Description			
Number of scanning lines in the image, also known as resolution height. For example, 1080, 720, 576 etc.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

scanningFormat

element (type)			
Description			
Whether the image is presented using interlaced or progressive scanning.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ScanningFormatType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

scanningOrder

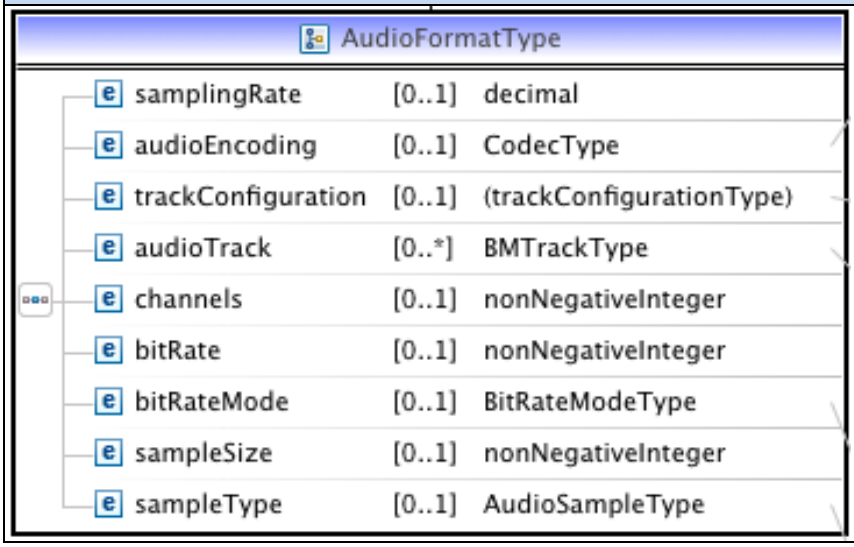
element (type)			
Description			
Whether the fields are ordered top (upper) or bottom (lower) field first in video stream encoding order.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ScanningOrderType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

noiseFilter

element (type)			
Description			
Whether noise was removed from the signal.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	boolean	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.9 AudioFormatType

complex type (base)		
Description		
Characteristics of an audio signal. See 'audioFormat' in EBU Tech 3293 for more information.		
Base	**Service Description**	**Content of Service Description**
bms:FormatType		
Normative Requirements		
Class Diagram		



samplingRate

element (type)			
Description			
Audio sampling rate that defines the number of samples per second taken from the continuous audio signal, expressed in Hertz.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	decimal	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

audioEncoding

element (type)			
Description			
Defines the audio compression format of the resource (e.g. AAC) for an audio channel.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:CodecType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

trackConfiguration

element (child, attributeGroup)			
Description			
Defines the audio track configuration that expresses the arrangement or audio tracks e.g. 'stereo', '2+1', 'surround', 'surround (7+1)' etc.			
Occurrence	**attributeGroup**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:typeGroup	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

audioTrack

element (type)			
Description			
Definition of audio tracks in the source.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	bms:BMTrackType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

channels

element (type)			
Description			
Number of channels used for the source recording.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

bitRate

element (type)			
Description			
Number of bits at which the audio is encoded, measured in bits per second.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

bitRateMode

element (type)			
Description			
Type of bitrate, either constant or variable.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:BitRateModeType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

sampleSize

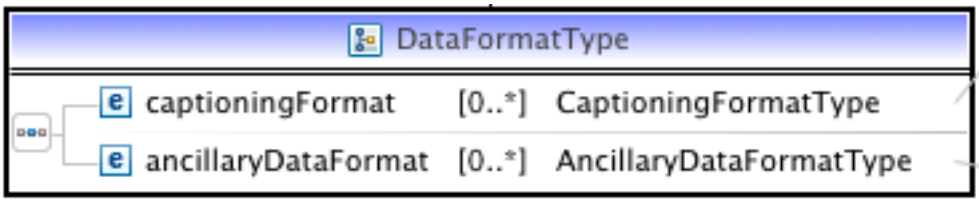
element (type)			
Description			
Sample size for the recording, e.g. 16- or 24-bits per sample.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

sampleType

element (type)			
Description			
Whether the samples are represented as integer or floating point values.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:AudioSampleType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.10 DataFormatType

complex type (base)		
Description		
Characteristics of a data signal, used to carry captioning or ancillary data.		
Base	**Service Description**	**Content of Service Description**
bms:FormatType		
Normative Requirements		
Class Diagram		



captioningFormat

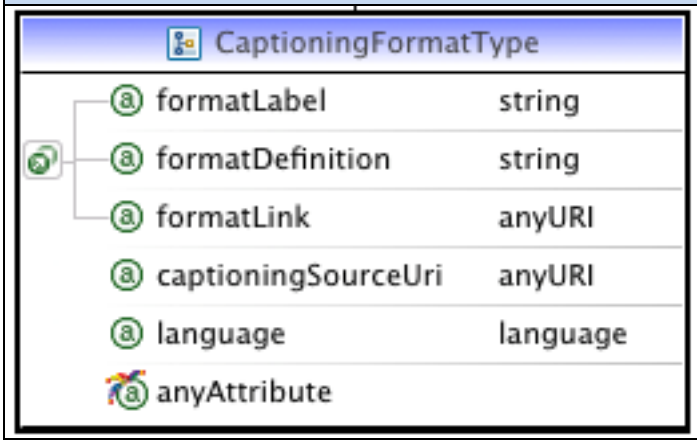
element (type)			
Description			
Where captioning data is present, describes the captioning formats and their purpose.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	bms:CaptioningFormatType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

ancillaryDataFormat

element (type)			
Description			
Where ancillary data is present, describes the captioning formats and their purpose.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	bms:AncillaryDataFormatType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.11 CaptioningFormatType

complex type (base)		
Description		
Describes a captioning format and its purpose.		
Base	**Service Description**	**Content of Service Description**
string		
Normative Requirements		
Class Diagram		



formatGroup

attributeGroup (ref)			
Description			
Underlying format for the captioning/subtitling being used.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1		Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

captioningSourceUri

attribute			
Description			
Location of the file with the captioning/subtitling data. See '@captioningSourceUri' in EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
optional	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

language

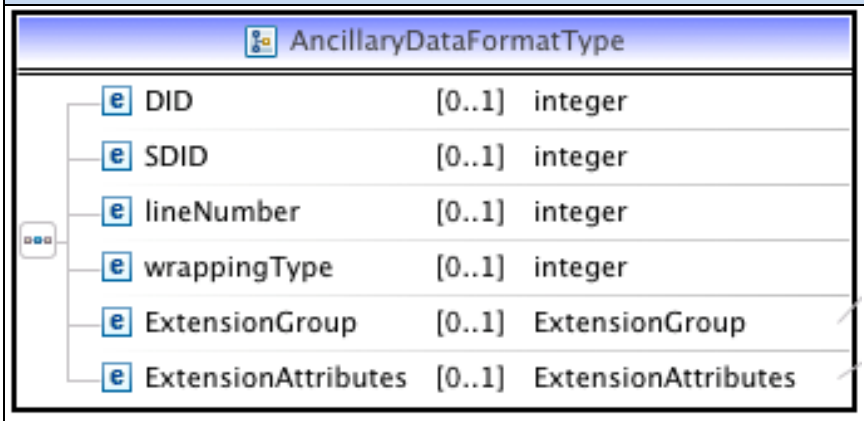
attribute			
Description			
Language of the caption as delivered (e.g., en-UK). See 'captioningFormat - dc:language' in EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
	language	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

anyAttribute

extension			
Description			
Attributes from other namespaces are permitted. Lax processing requests that a validator checks the attributes where access to the attribute's defining schema is available.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
	any	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			
Attributes from the FIMS base namespace shall not be used.			

S0 2.3.12 AncillaryDataFormatType

complex type	
Description	
Ancillary data packet type. See SMPTE ST 291 and SMPTE ST 436.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



DID

element (type)			
Description			
ANC DID value.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	integer	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

SDID

element (type)			
Description			
ANC SDID value.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	integer	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

lineNumber

element (type)			
Description			
Video line number containing the ANC packets of this type.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	integer	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

wrappingType

element (type)			
Description			
Indicates HANC or VANC, and what field in which packets should be stored. See SMPTE ST 436 for legal values.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	integer	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

ExtensionGroup

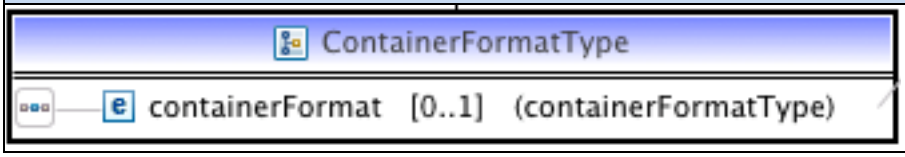
element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup		
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes		
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.13 ContainerFormatType

complex type (base)		
Description		
Container/wrapper format that is used in complement to the streams encoding, for example MXF, wave, Quicktime, etc. See 'fileFormat' in EBU Tech 3293 for more information.		
Base	**Service Description**	**Content of Service Description**
bms:FormatType		
Normative Requirements		
Class Diagram		



containerFormat

element (child, base)				
Description				
Defines the file format for the data. See 'fileFormat' in EBU Tech 3293 for more information.				
Occurrence	**Base**	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	bms:formatGroup	Optional	Optional
Service Description	**Contents of Service Description**			
Required	Whether on request, it must be present, may be interpreted or is not applicable.			
Normative Requirements				

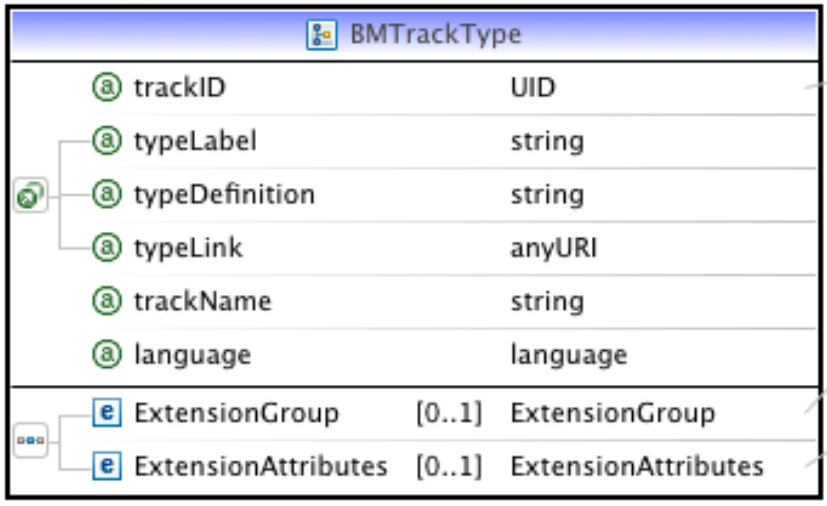
S0 2.3.14 BMTrackType

complex type
Description
Tracks expose the underlying structural metadata of the content streams embedded inside a physical content essence. Examples of tracks are the separate audio and video streams inside an audiovisual content essence. Tracks have a category, for example "main" (videoTrack), "audioDescription" (audioTrack), "closed captioning" (dataTrack). Tracks have also an identifier and a description.

Service Description	Content of Service Description

Normative Requirements

Class Diagram



ExtensionGroup

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup		
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes		
Service Description	**Contents of Service Description**		
Normative Requirements			

trackID

attribute			
Description			
Identifier of the track.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	bms:UID	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

bms:typeGroup

attributeGroup (ref)			
Description			
The category of the track (e.g. "video").			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
		Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

trackName

element (type)			
Description			
Name for the track.			
Occurrence/Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

language

attribute			
Description			
Language of the track, such as the primary spoken language for an audio track.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	language	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

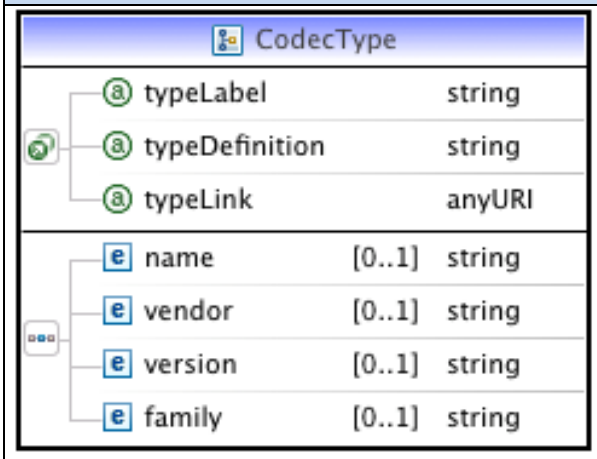
S0 2.3.15 CodecType

complex type
Description
Describes a codec used for video or audio encoding.

Service Description	Content of Service Description

Normative Requirements

Class Diagram



name

element (type)			
Description			
Name of the hardware or software codec.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

vendor

element (type)			
Description			
The vendor/organization that produced the codec.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

version

element (type)			
Description			
Version of the specific release of the codec.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

family

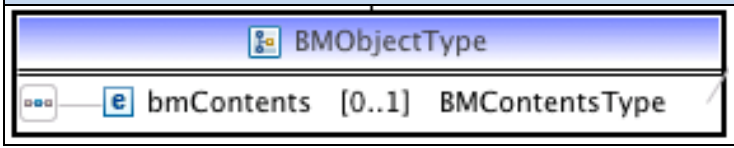
element (type)			
Description			
Codec family, e.g. MPEG-2, H.264 etc.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

bms:typeGroup

attributeGroup (ref)			
Description			
To define the encoding format either as a string using typeLabel or as a pointer to a classification scheme using typeLink. An optional definition can be provided using the typeDefinition.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
		Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.16 BMObjectType

complex type (base)		
Description		
Common representation of the content exchanged by FIMS media services, through reference to logical content objects. Note that although the current BMObject can only reference at most one logical content item, it is intended that future versions of FIMS will extend BMObject to provide different kinds of content collections, such as sequences and edit decision lists.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Class Diagram		

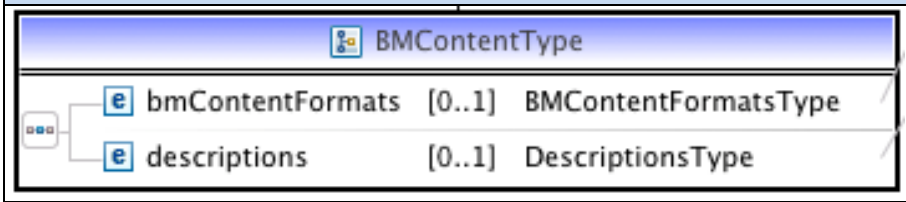


bmContents

element (type)			
Description			
Logical content item that is exchanged by a FIMS media service.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	BMContentsType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.17 BMContentType

complex type (base)		
Description		
Logical items of content, representing a sequence of frames and/or samples that have a fixed length and are intended to be played sequentially from end-to-end.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Logical content items should be used to representing a single version of a programme or clip that has many different renditions, e.g. different encodings. Logical content items shall not be used to represent different editorial versions of content, such as a different cut of a programme created for a different audience.		
Class Diagram		



bmContentFormats

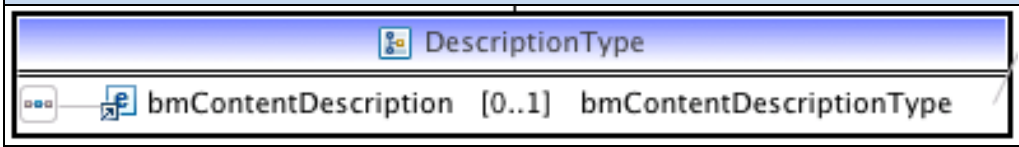
element (type)			
Description			
List of different content formats that provide physical representations of the logical content item. Content formats are different renditions of the same content, differing in the way they are encoded. Content formats have technical metadata.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:BMContentFormatsType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			
All content formats shall have the same play time, with the exception of minor alterations due to frame rate conversions.			

descriptions

element (type)			
Description			
Non-technical descriptions of the logical content. It is recommended that a basic description of the content items is included by making use of the FIMS description metadata schema (http://description.fims.tv), which is based on a subset of descriptive metadata fields from EBUCore and Dublin Core. This will be useful for interoperable monitoring applications by providing a consistent place where, say, a title field can be found for presentation in a GUI. Content items may have many different descriptions for different purposes, such as for archiving, presentation, rights management and contributor management. For this purpose, the use of multiple descriptions in included. Also, a programme may be an episode that is part of a series that is part of a brand and the descriptions of each level in that hierarchy could be included (by reference) here.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:DescriptionsType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			
At least one description of a logical content item should be provided. One of the descriptions should include an element containing the basic metadata described according to the FIMS description scheme (http://description.fims.tv).			

S0 2.3.18 DescriptionType

complex type (base)		
Description		
Description of a logical content, containing descriptive metadata. Descriptive metadata is attached using the XML schema "any" facility supported by all resources.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Class Diagram		

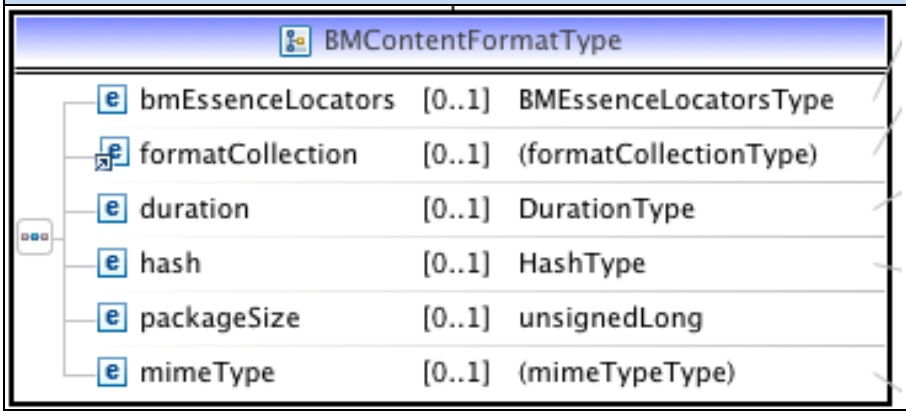


desc:bmContentDescription

element (ref)			
Description			
Description of a logical content, containing core descriptive metadata based on EBUCore.			
Occurrence	**Ref**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	desc:bmContentDescription	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.3.19 BMContentFormatType

complex type (base)		
Description		
Format description and essence locations of a physical representation of single rendition of a logical content item.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
Class Diagram		



bmEssenceLocators

element (type)			
Description			
Known locations for the content.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:BMEssenceLocatorsType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			
All the locations described shall store the same content and shall share the format description that is described by the formatCollection property of this content format.			

bms:formatCollection

element (ref)			
Description			
Technical metadata describing this specific physical embodiment of the content. The description contains the video, audio, data and container formats of the content format.			
Occurrence	**Ref**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:formatCollection	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

duration

element (type)			
Description			
Duration of the audio and/or visual content.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:DurationType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

hash

element (type)			
Description			
Hash code value calculated for the physical essence.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:HashType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			
If the essence location is a collection of files rather than a single file and the hash is expected to be produced by a service, then the service shall describe the algorithm used to calculate a hash value for multiple files.			

packageSize

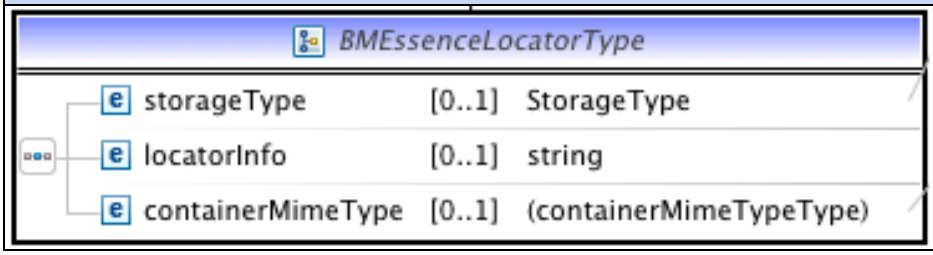
element (type)			
Description			
Size of the file or collection of files that provide the physical essence representation.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	unsignedLong	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			
If the essence representation is a single file, this value shall be set and be equal to the file size. If the package is a collection of files, rather than a single file, then the packageSize shall correspond to the sum of the sizes of the different components of the package.			

mimeType

element (child, base)				
Description				
Multipurpose Internet Mail Extensions Media Type for the physical essence representation. The MIME type can include any required or optional parameters after the type, separated by semicolons. Note that this MIME type refers to the overall type of media, for example a folder containing P2 card data or an AS-02 bundle both use MXF file formats, setting this property to "application/mxf".				
Occurrence	**Base**	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	bms:typeGroup	Optional	Optional
Service Description	**Contents of Service Description**			
Required	Whether on request, it must be present, may be interpreted or is not applicable.			
Normative Requirements				
The format of the media type shall be as defined in IETF RFC 2046 and the MIME type should be registered by IANA at http://www.iana.org/assignments/media-types/index.html.				

S0 2.3.20 BMEssenceLocatorType

complex type (abstract, base)		
Description		
Location of a physical representation of an item of content. Note that essence locations may be shared by multiple physical content formats and/or logical content items. For example, a folder structure taken from a camera may contain both master-quality and proxy-quality versions. The same essence locator can be targeted by two different physical content formats for the same logical content item. As another example, an AS-02 bundle may contain two different editorial versions of the same programme and can be referenced from two separate logical content items.		
Base	**Service Description**	**Content of Service Description**
bms:ResourceType		
Normative Requirements		
An essence locator may be the target of more than one references from a content format.		
Class Diagram		



storageType

element (type)			
Description			
Kind of storage for the essence, for example "online" or "hsm".			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	StorageType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

locatorinfo

element (type)			
Description			
Describes the geographical location of the content, for example "New York".			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

containerMimeType

element (child, base)				
Description				
Multipurpose Internet Mail Extensions Media Type for the physical container used for the representation. Vendors are encouraged to register MIME types for the containers in use, such as a folder containing capture data from a camera. Where additional structural data is required, vendors are encouraged to extend one of the essence locator types (single file, list, folder).				
Occurrence	**Base**	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	bms:typeGroup	Optional	Optional
Service Description	**Contents of Service Description**			
Required	Whether on request, it must be present, may be interpreted or is not applicable.			
Normative Requirements				
The format of the media type shall be as defined in IETF RFC 2046 and the MIME type should be registered by IANA at http://www.iana.org/assignments/media-types/index.html.				

S0 2.3.21 SimpleFileLocatorType

complex type (base)		
Description		
Location of essence represented by a single file		
Base	**Service Description**	**Content of Service Description**
bms:BMEssenceLocatorType		
Normative Requirements		
Class Diagram		

file

element (type)			
Description			
Location of the essence-containing single file, such as an MXF OP-1a file.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.22 ListFileLocatorType

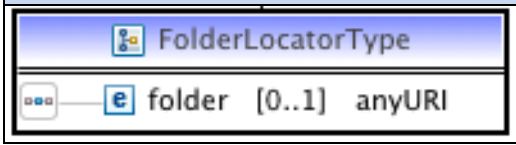
complex type (base)		
Description		
Location of essence represented by a list of files that represent a single rendition of the content.		
Base	**Service Description**	**Content of Service Description**
bms:BMEssenceLocatorType		
Normative Requirements		
Class Diagram		

file

element (type)			
Description			
List of a collection of files that, taken together, represent an essence location. Note that the files are not necessarily stored in the folder, on the same disk resource or even on the same server.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.3.23 FolderLocatorType

complex type (base)		
Description		
Root of a folder structure that, taken as a whole including its children, represents an essence location that stores the rendition of the content described by the parent content format.		
Base	**Service Description**	**Content of Service Description**
bms:BMEssenceLocatorType		
Normative Requirements		
The folder and all of its children should be presented within a single filing system.		
Class Diagram		



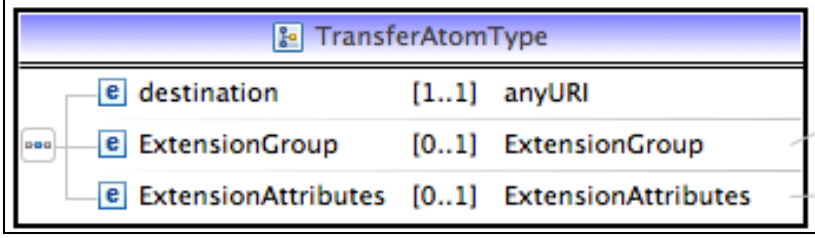
folder

element (type)			
Description			
Root of a folder structure where the root and its children are an essence representation.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Whether on request, it must be present, may be interpreted or is not applicable.		
Normative Requirements			

S0 2.4 FIMS-defined atoms

S0 2.4.1 TransferAtomType

complex type	
Description	
Parameters specific to transfer media services that can be re-used for other services. Note: Some parameters might be added to the atom in future version to specify network resource utilization, a list of acceptable transfer mechanisms or additional security options including whether the received files should be checked against a fingerprint and whether they are encrypted.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



destination

element (type)			
Description			
Destination path for the target.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	anyURI	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Required	Kinds of destination supported.		
Normative Requirements			
The URL field shall follow the standard structure for a URL as defined in RFC1738 – Uniform Resource Locators (URL). In case of a file protocol scheme, the host field shall explicitly be included in the URL. The URL covers different protocol schemes, such as NFS mounts RFC2224, which defines nfs://[host]:[port]/[directory]...[filename], and the internet draft for the smb: scheme for CIFS, with the restriction that only the path and not the file name shall be provided.			

ExtensionGroup

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup		
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes		
Service Description	**Contents of Service Description**		
Normative Requirements			

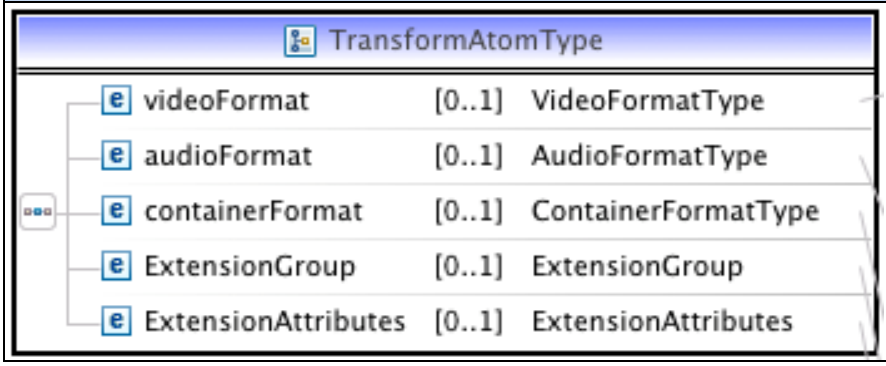
S0 2.4.2 TransformAtomType

complex type
Description
Parameters specific to the transform media service that can be re-used by other services. Note: Some parameters might be added to this type in future version of this specification to specify AV Process, etc.

Service Description	Content of Service Description

Normative Requirements

Class Diagram



videoFormat

element (type)			
Description			
Specifies the output video format. See 'videoFormat' in EBU tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:VideoFormatType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not. If supported, the video formats that are supported.		
Normative Requirements			

audioFormat

element (type)			
Description			
Specifies the output audio format. See 'audioFormat' in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:AudioFormatType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not. If supported, the audio formats that are supported.		
Normative Requirements			

containerFormat

element (type)			
Description			
Container/wrapper format of the output file. See "fileFormat" in EBU Tech 3293 for more information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ContainerFormatType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not. If supported, the container formats that are supported.		
Normative Requirements			

ExtensionGroup

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup		
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes		
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.5 FIMS-defined attribute groups

S0 2.5.1 typeGroup

attributeGroup	
Description	
Defines a contextual type.	
Service Description	**Contents of Service Description**
Normative Requirements	

typeLabel

attribute			
Description			
Free text. Example: 'H264 Main Profile @ Level 1'. See EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

typeDefinition

attribute			
Description			
Free text for an optional definition. Example: 'the video compression scheme H264, main profile, level1 as specified by ISO/IEC'. See EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

typeLink

attribute			
Description			
Link to a classification scheme. See EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.5.2 formatGroup

attributeGroup	
Description	
Defines a format.	
Service Description	**Contents of Service Description**
Normative Requirements	

formatLabel

attribute			
Description			
Free text field. See EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

formatDefinition

attribute			
Description			
An optional definition. See EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

formatLink

attribute			
Description			
An optional definition. See EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	anyURI	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.6 FIMS-defined simple types

S0 2.6.1 UID

simple type (union)
Description
Unique identifiers that are based on SMPTE 2029, providing a generic representation for UUIDs, ULs and UMIDs.
Union
bms:UUID, bms:UMID, bms:UL, bms:EmptyID
Normative Requirements
Although empty IDs are permitted, these should only be used for initial requests where the server is expected to create an identifier.

S0 2.6.2 UUID

simple type (base)	
Description	
A universally unique identifier (UUID) or globally unique identifier (GUID) that may be represented as a URN as per RFC 4122.	
restriction base	**restriction**
string	pattern = (urn:uuid:)?[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}
Normative Requirements	

S0 2.6.3 UMID

simple type (base)	
Description	
A SMPTE unique material number, as specified in SMPTE ST 330 and represented as a URN according to SMPTE ST 2029.	
restriction base	**pattern**
string	pattern = urn:smpte:umid:([0-9a-fA-F]{8}\.){7}[0-9a-fA-F]{8}
Normative Requirements	

S0 2.6.4 UL

simple type (base)	
Description	
A SMPTE universal label, as specified in SMPTE ST 298 and represented as a URN according to SMPTE ST 2029.	
restriction base	**restriction**
string	pattern = urn:smpte:ul:([0-9a-fA-F]{8}\.){3}[0-9a-fA-F]{8}
Normative Requirements	

S0 2.6.5 EmptyID

simple type (base)	
Description	
Empty identification permitted to enable validation of documents where identities are not yet known.	
restriction base	**restriction**
string	length = 0
Normative Requirements	

S0 2.6.6 ResourceIDType

simple type (base)	
Description	
This type provides a unique way to identify a resource.	
restriction base	**restriction**
UID	
Normative Requirements	

S0 2.6.7 RevisionIDType

simple type (base)	
Description	
Identification of a specific revision.	
restriction base	**restriction**
NMTOKEN	
Normative Requirements	

S0 2.6.8 Timecode

simple type (base)	
Description	
A SMPTE ST 12-1 timecode value formatted according to the edit decision list grammar specified in SMPTE ST 258	
restriction base	**restriction**
string	pattern = (([0-1][0-9])|([2][0-3])):[0-5][0-9]:[0-5][0-9](([.,])|([:;]))[0-2][0-9]
Normative Requirements	

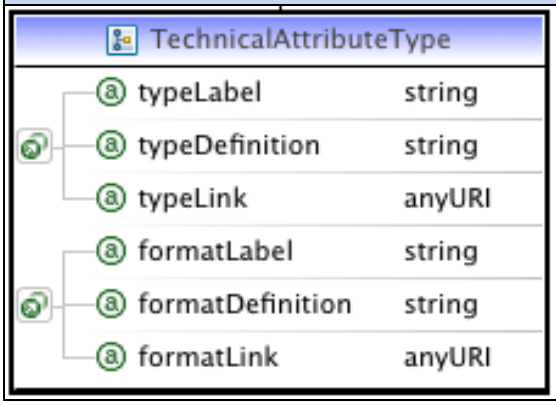
S0 2.6.9 TimecodeDuration

simple type (base)	
Description	
A representation of AV duration as a SMPTE timecode-like value, with an upper bound of 99 hours rather than 23 hours.	
restriction base	**restriction**
string	pattern = [0-9][0-9]:[0-5][0-9]:[0-5][0-9](([.,])|([:;]))[0-2][0-9]
Normative Requirements	

S0 2.7 FIMS-defined complex types for representing single values

S0 2.7.1 TechnicalAttributeType

complex type (base)		
Description		
Allows users / implementers to define their own technical parameters as 'string' for which a format can be defined to restrict the string format. See 'TechnicalAttributeString' in EBU Tech 3293 for more information.		
Base	**Service Description**	**Content of Service Description**
string		
Normative Requirements		
Class Diagram		



bms:typeGroup

attributeGroup (ref)			
Description			
Type information for the technical attribute, as specified in EBU Tech 3293			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
		Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

bms:formatGroup

attributeGroup (ref)			
Description			
Format information for the technical attribute, as specified in EBU Tech 3293.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
		Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.2 LengthType

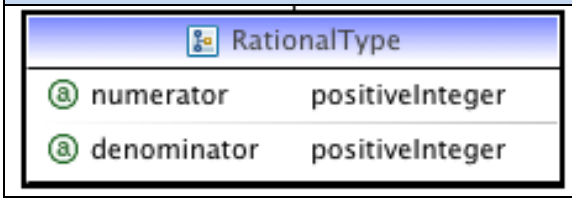
complex type (base)		
Description		
A length value and its unit of measurement.		
Base	**Service Description**	**Content of Service Description**
nonNegativeInteger		
Normative Requirements		
Class Diagram		

unit

attribute			
Description			
Specifies the unit in which the length is expressed. See '@unit' in EBU Tech 3293 for more information.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional	string	Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.3 RationalType

complex type (base)		
Description		
A rational value expressed by its fraction of a second numerator and denominator components. Rational values can be used to represent edit rates. For example, a frame rate of 29.97 would be represented as 30 corrected by 1000 (numerator) / 1001 (denominator).		
Base	**Service Description**	**Content of Service Description**
long		
Normative Requirements		
Class Diagram		



numerator

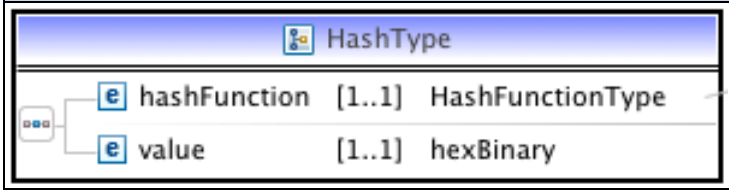
attribute			
Description			
The numerator of the rational value.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
required	positiveInteger	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

denominator

attribute			
Description			
The denominator of the rational value.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
required	positiveInteger	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.4 HashType

complex type	
Description	
Provides information on the algorithm used in an integrity check process. It is based on the 'hash' type defined in SMPTE ST 2032.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



hashFunction

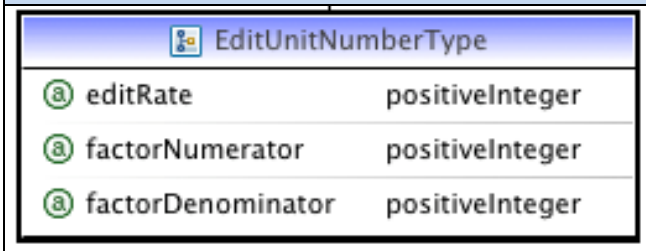
element (type)			
Description			
The hash function used to hash the content (e.g. CRC32).			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:HashFunctionTyp	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

value

element (type)			
Description			
Hash (digest) value as a hexadecimal string.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	hexBinary	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.5 EditUnitNumberType

complex type (base)		
Description		
A number of edit units as defined in EBU Tech 3293. An editUnit is the inverse of the edit rate, or corrected edit rate as the result of editUnit=1/(editRate*(factorNumerator/factorDenominator)). See EBU Tech 3293 for more information. Values of the combination of editRate, factorNumerator, and factorDenominator shall be subject to the constraints listed in the Annex EditUnitNumberType.		
Base	**Service Description**	**Content of Service Description**
long		
Normative Requirements		
Class Diagram		



editRate

attribute			
Description			
The base number of frames or samples per seconds. This base number can be corrected by a factor calculated as the product of the edit rate and 'factorNumerator/factorDenominator'.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
required	positiveInteger	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

factorNumerator

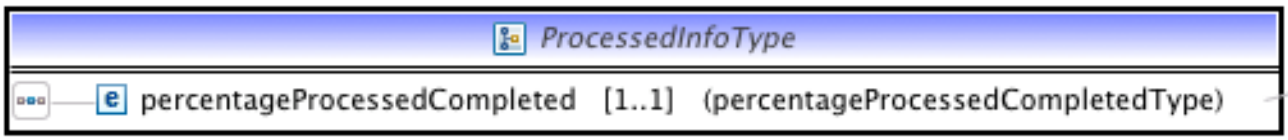
attribute			
Description			
The numerator of the correction factor.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
required	positiveInteger	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

factorDenominator

attribute			
Description			
The denominator of the correction factor.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
required	positiveInteger	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.6 ProcessedInfoType

complex type (abstract)	
Description	
Statistics on the amount of information processed so far. For example, in terms of bytes or frames processed.	
Service Description	**Content of Service Description**
Required	The type of processed information provided.
Normative Requirements	
Class Diagram	



percentageProcessedCompleted

element (type)			
Description			
The percentage of job completed. The percentage can be based on one of two metrics: number of bytes processed, or number of frames processed (e.g. for transcoding video files). If the "processedBytesCount" metric is provided, then the percentage is based on number of bytes processed. If the "processedFramesCount" metric is provided, then the percentage is based on the number of processed frames.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	nonNegativeInteger (0-100)	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.7 ProcessedInfoByBytesType

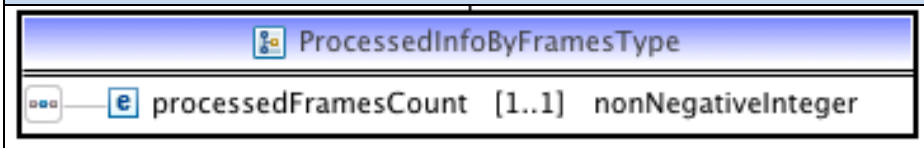
complex type (base)		
Description		
Statistics on number of bytes processed.		
Base	**Service Description**	**Content of Service Description**
bms:ProcessedInfoType		
Normative Requirements		
Class Diagram		

processedBytesCount

element (type)			
Description			
The number of bytes processed from the start of the job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	unsignedLong	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.8 ProcessedInfoByFramesType

complex type (base)		
Description		
Statistics on number of frames processed.		
Base	**Service Description**	**Content of Service Description**
bms:ProcessedInfoType		
Normative Requirements		
Class Diagram		

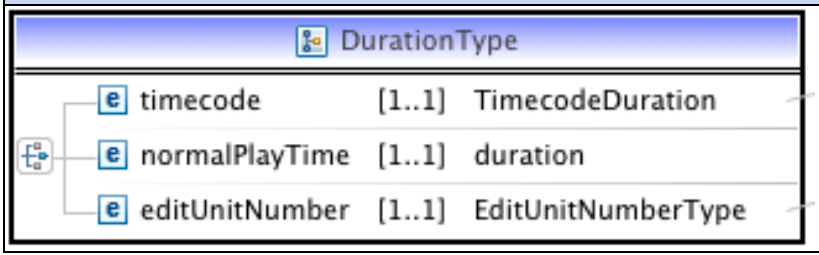


processedFramesCount

element (type)			
Description			
The number of frames (for audiovisual content only) processed from the beginning of the job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	nonNegativeInteger	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.9 DurationType

complex type	
Description	
Measure of the duration of an operation or item of content.	
Service Description	**Content of Service Description**
Required	Element(s) supported.
Normative Requirements	
One of the elements shall be used.	
Class Diagram	



timecode

element (type)			
Description			
Duration of AV content in a timecode-like format (e.g. SMPTE ST 12-1) as defined in EBU Tech 3293.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:TimecodeDuration	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			
This kind of duration shall be used only where an exact time can be calculated from the timecode value. For example, it is not appropriate for NTSC systems.			

normalPlayTime

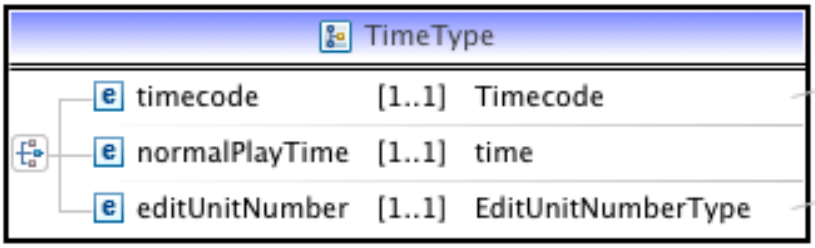
element (type)			
Description			
A duration expressed using usual time representation as in RFC 2326 and ISO 8601 (e.g.: PT1H31M25.152S)			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	duration	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			
This kind of duration shall be used only where an exact time can be calculated from the timecode value. For example, it is not appropriate for NTSC systems.			

editUnitNumber

element (type)			
Description			
A duration expressed as a number of edit units, as defined in EBU Tech 3293.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:EditUnitNumberType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.7.10 TimeType

complex type	
Description	
Value used to represent a point in time, such as at what time to start an operation.	
Service Description	**Content of Service Description**
Required	Element(s) supported.
Normative Requirements	
One of the elements shall be used.	
Class Diagram	



timecode

element (type)			
Description			
Start time expressed using timecode values compliant with SMPTE ST 12-1.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:Timecode	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

normalPlayTime

element (type)			
Description			
Start time expressed by time of day in the format hh:mm:ss.sss with an optional following time zone indicator.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	time	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

editUnitNumber

element (type)			
Description			
Start time expressed as a number of edit units.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:EditUnitNumberType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.8 FIMS-defined enumerations

S0 2.8.1 QueueStatusType

simple type (enum)	
Description	
Current state of a queue.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
started	Queue has started successfully.
stopped	Queue has been stopped successfully. The queue is not accepting newly submitted jobs.
locked	Queue has been locked successfully and will not accept any new submitted jobs. The jobs already in the queue will continue to start or run to completion.

S0 2.8.2 PriorityType

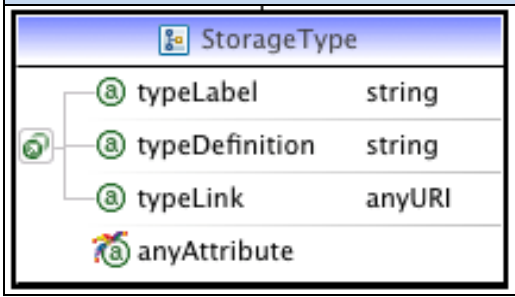
simple type (enum)	
Description	
Acceptable priority values used to determine the execution order of jobs. See specification Part 1 'Job Execution Priority'.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
low	Job initially allocated to the end of the queue.
medium	Job initially allocated to be executed before any low priority jobs but after any existing medium priority jobs.
high	Job initially allocated before any medium and low priority jobs but after existing high priority jobs.
urgent	Job initially allocated to be executed before any high, medium and low priority jobs but after existing urgent jobs.
immediate	Job should be executed as soon as the request is received.

S0 2.8.3 StorageTypes

simple type (enum)	
Description	
Different kinds of storage media available (e.g. online on disk).	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
online	
offline	
hsm	
archive	
playout	
other	

S0 2.8.4 StorageType

complex type (base)		
Description		
Different kinds of storage media available (e.g. online on disk), in combination with type group attributes.		
Base	**Service Description**	**Content of Service Description**
bms:StorageTypes		
Normative Requirements		
Class Diagram		



bms:typeGroup

attributeGroup (ref)			
Description			
Storage type.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Service Description	**Contents of Service Description**		
Normative Requirements			

anyAttribute

anyAttribute			
Description			
Attributes from other namespaces are permitted. Lax processing requests that a validator checks the attributes where access to the attribute's defining schema is available.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Service Description	**Contents of Service Description**		
Normative Requirements			
Attributes from the FIMS base namespace shall not be used.			

S0 2.8.5 QueueCommandType

simple type (enum)	
Description	
Commands for the control and management of a queue. All commands are optional.	
Service Description	**Content of Service Description**
Required	Supported commands
Normative Requirements	
Enum Value	**Enum Description**
status	Retrieve the current status of the queue.
clear	Delete all remaining jobs in the queue.
stop	Stop the queue. Jobs cannot then be en-queued or de-queued.
start	Restart a stopped queue.
lock	Lock the queue. Jobs cannot be en-queued but they are still being processed and can be deleted.
unlock	Unlock a locked queue.

S0 2.8.6 JobCommandType

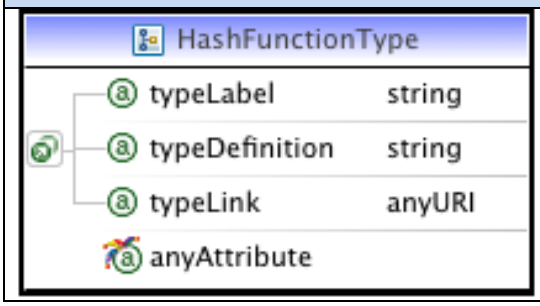
simple type (enum)	
Description	
Commands for the control and management of a job. All commands are optional.	
Service Description	**Content of Service Description**
Required	Supported commands
Normative Requirements	
Enum Value	**Enum Description**
cancel	Cancel the job.
pause	Pause the job. It can be restarted with resume.
resume	Resume the job from its paused state.
restart	Restart the job from the beginning.
stop	Stop the job.
cleanup	Remove all the data associated with the job.
modifyPriority	Modify the priority of the job.

S0 2.8.7 HashFunctionTypes

simple type (enum)	
Description	
Different hash functions that can be used to hash the content, as defined in SMPTE ST 2032.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
CRC32	32-bit Cyclic Redundancy Check, as defined in ISO 3309.
CRC64	64-bit Cyclic Redundancy Check, as defined in ISO 3309.
MD5	128-bit Message Digest Algorithm 5, as defined in RFC 1321.
SHA1	160-bit Secure Hash Algorithm, as defined in FIPS 180-2.
SHA256	256-bit Secure Hash Algorithm, as defined in FIPS 180-2.
SHA384	84-bit Secure Hash Algorithm, as defined in FIPS 180-2.
SHA512	512-bit Secure Hash Algorithm, as defined in FIPS 180-2.

S0 2.8.8 HashFunctionType

complex type (base)		
Description		
Different hash functions that can be used to hash the content, as defined in SMPTE ST 2032, in combination with type group description.		
Base	**Service Description**	**Content of Service Description**
bms:HashFunctionTypes		
Normative Requirements		
Class Diagram		



bms:typeGroup

attributeGroup (ref)			
Description			
Hash function type.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Service Description	**Contents of Service Description**		
Normative Requirements			

anyAttribute

anyAttribute			
Description			
Attributes from other namespaces are permitted. Lax processing requests that a validator checks the attributes where access to the attribute's defining schema is available.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Service Description	**Contents of Service Description**		
Normative Requirements			
Attributes from the FIMS base namespace shall not be used.			

S0 2.8.9 JobStatusType

simple type (enum)	
Description	
All possible states for a job.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
queued	Job is in the queue and is ready to start.
running	Job has started successfully and is currently running.
paused	Job has been paused.
completed	Job has completed successfully.
canceled	Job has been cancelled.
stopped	Job has been stopped. A partial result may be retrieved.
failed	Job has ended with an error.
cleaned	A cleanup command was issued for the job. All data related to the job have been removed.
unknown	Status for the job is unknown.

S0 2.8.10 AudioSampleType

simple type (enum)	
Description	
Audio sample representation: integer or floating point values.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
float	
integer	

S0 2.8.11 ScanningFormatType

simple type (enum)	
Description	
Scanning format representation: interlaced or progressive.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
interlaced	
progressive	

S0 2.8.12 ScanningOrderType

simple type (enum)	
Description	
Scanning order representation: whether the fields are ordered top (upper) or bottom (lower) field first.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
top	
bottom	

S0 2.8.13 BitRateModeType

simple type (enum)	
Description	
Representation of type of bitrate: constant or variable.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
constant	
variable	

S0 2.8.14 ErrorCodeType

simple type (enum)	
Description	
Common error codes which can be shared by different classes of adapters. Error codes are classified in five main categories: • INF_S00_xxxx: Infrastructure errors (system, storage, network, memory, processor) • DAT_S00_xxxx: Data errors (validation, missing, duplication) • SVC_S00_xxxx: Operation errors (existence, support, lock, connection, failure) • SEC_S00_xxxx: Security errors (authentication, authorization) • EXT_S00_xxxx: Extended code. See extended error code for detail.	
Service Description	**Content of Service Description**
Normative Requirements	

Enum Value	Enum Description
INF_S00_0001	System unavailable.
INF_S00_0002	System timeout.
INF_S00_0003	System internal error.
INF_S00_0004	Unable to connect to the database.
INF_S00_0005	System out of memory.
INF_S00_0006	System out of disk space.
SVC_S00_0001	Job Command is not currently supported by the service URI specified
SVC_S00_0002	Queue Command is not currently supported by the service or the device.
SVC_S00_0003	Operation requested is not currently supported by the service or the device.
SVC_S00_0004	Service unable to find/lookup device endpoint.
SVC_S00_0005	Job command failed.
SVC_S00_0006	Queue command failed.
SVC_S00_0007	Service unable to connect to device endpoint
SVC_S00_0008	Job queue is full, locked or stopped. No new jobs are being accepted.
SVC_S00_0009	Job ended with a failure.
SVC_S00_0010	Service received no response from device.
SVC_S00_0011	Service received an exception from device. See description or exception detail.
SVC_S00_0012	Service received an unknown or an internal error from device. See description for error detail.
SVC_S00_0013	Unable to connect to client's notification service endpoint (replyTo) to send the asynchronous job result notification response.
SVC_S00_0014	Unable to connect to client's service endpoint (faultTo) to send the asynchronous job fault response.
SVC_S00_0015	Feature not supported.
SVC_S00_0016	Deadline passed.
SVC_S00_0017	Time constraints in request cannot be met.
SVC_S00_0018	Internal or Unknown error encountered. See description for error detail.
DAT_S00_0001	Invalid request, XML format.
DAT_S00_0002	Invalid input media format.
DAT_S00_0003	Invalid jobID - the supplied jobID does not exists.
DAT_S00_0004	Missing required service metadata in request.
DAT_S00_0005	Duplicate jobGUID detected for new job.
DAT_S00_0006	Invalid request parameters.
DAT_S00_0007	Job command not valid.
DAT_S00_0008	Queue command not valid.
DAT_S00_0009	Invalid priority.
DAT_S00_0010	Input media not found. Invalid resource URI specified.
EXT_S00_0000	Extended code. See extended error code for details.

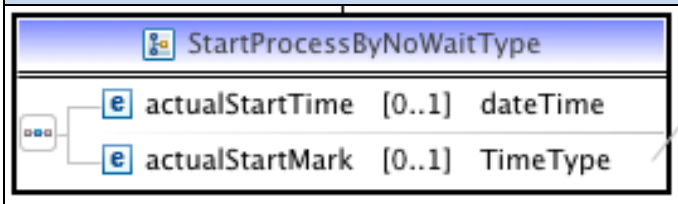
S0 2.9 FIMS-defined control data types

S0 2.9.1 StartProcessType

complex type (abstract)	
Description	
Kinds of time, time code, or event based information used to start a real-time process such as a capture process.	
Service Description	**Content of Service Description**
Required	Supported start process types.
Normative Requirements	
One of the following types shall be used. StartProcessByNoWaitType and StartProcessByServiceDefinedTimeType shall be supported.	
Class Diagram	
	

S0 2.9.2 StartProcessByNoWaitType

complex type (base)		
Description		
Start a real-time process with no initial wait.		
Base	**Service Description**	**Content of Service Description**
bms:StartProcessType		
Normative Requirements		
The process shall start without waiting for a specific time or event.		
Class Diagram		



actualStartTime

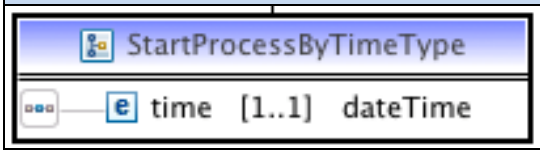
element (type)			
Description			
Actual time that the process started.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
If supported, the value shall be provided on completion of the process.			

actualStartMark

element (type)			
Description			
Time mark indicating the time that the process started as a value measured along the associated audio and/or video stream.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:TimeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

S0 2.9.3 StartProcessByTimeType

complex type (base)		
Description		
Provides the time when a real-time process should start.		
Base	**Service Description**	**Content of Service Description**
bms:StartProcessType		
Normative Requirements		
Class Diagram		

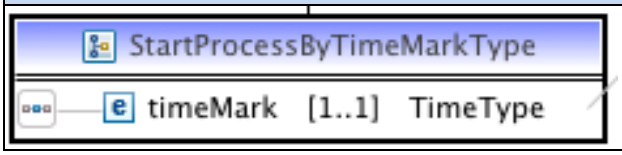


time

element (type)			
Description			
The time for starting a real-time process.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	dateTime	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			
Date and time information shall be set with format defined according to the dateTime type and shall include the timezone component.			

S0 2.9.4 StartProcessByTimeMarkType

complex type (base)		
Description		
Provides a video or audio stream time mark value to indicate the required start of a real-time process.		
Base	**Service Description**	**Content of Service Description**
bms:StartProcessType		
Normative Requirements		
Class Diagram		

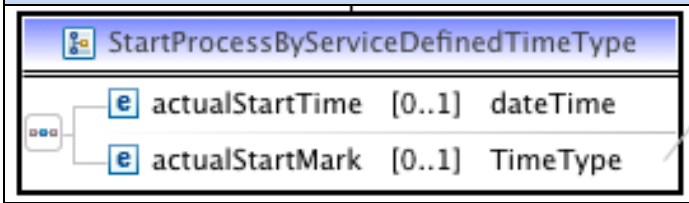


timeMark

element (type)			
Description			
A video and audio time point reference.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	TimeType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Required	Kinds of time mark the service supports.		
Normative Requirements			
The time reference shall be defined according to EBU Tech 3295. For example, SMPTE ST 12-1 time code.			

S0 2.9.5 StartProcessByServiceDefinedTimeType

complex type (base)		
Description		
A service-specific time mechanism specifies when a process is required to start.		
Base	**Service Description**	**Content of Service Description**
bms:StartProcessType		
Normative Requirements		
The service shall wait for an event to start the process. The actual event depends on the service implementation. A service shall support this type.		
Class Diagram		



actualStartTime

element (type)			
Description			
Actual time that the process started.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
If supported, the value shall be provided on completion of the process.			

actualStartMark

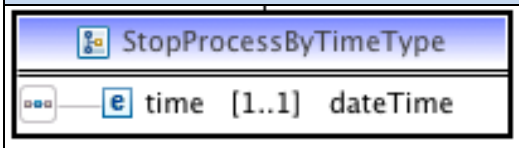
element (type)			
Description			
Time mark indicating the time that the process started as a value measured along the associated audio and/or video stream.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	TimeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not		
Normative Requirements			

S0 2.9.6 StopProcessType

complex type (abstract)	
Description	
Kind of time, time code, or event information used to stop a real-time process, such as a capture process.	
Service Description	**Content of Service Description**
Required	Supported stop process types.
Normative Requirements	
One of the following types shall be used. StopProcessByServiceDefinedTimeType shall be supported.	
Class Diagram	
	

S0 2.9.7 StopProcessByTimeType

complex type (base)		
Description		
The time when a real-time process should stop.		
Base	**Service Description**	**Content of Service Description**
bms:StopProcessType		
Normative Requirements		
Class Diagram		

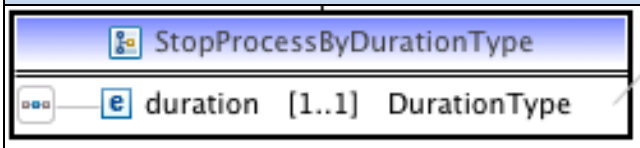


time

element (type)			
Description			
Time to stop a real-time process.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	dateTime	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			
Date and time information shall be set with format defined according to the dateTime type including the timezone			

S0 2.9.8 StopProcessByDurationType

complex type (base)		
Description		
Total duration of the real-time process.		
Base	**Service Description**	**Content of Service Description**
bms:StopProcessType		
Normative Requirements		
A real time process shall stop once the specified duration has elapsed.		
Class Diagram		

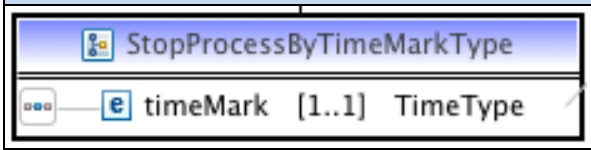


duration

element (type)			
Description			
Total required duration of the real-time process.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:DurationType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Required	Kind of duration supported		
Normative Requirements			

S0 2.9.9 StopProcessByTimeMarkType

complex type (base)		
Description		
A video or audio stream time reference when the real-time process stops.		
Base	**Service Description**	**Content of Service Description**
bms:StopProcessType		
Normative Requirements		
Class Diagram		

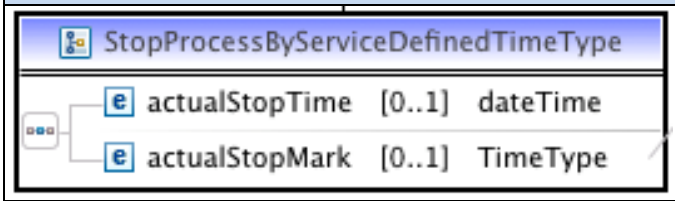


timeMark

element (type)			
Description			
A video or audio time point reference.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:TimeType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Required	Kind of time mark supported		
Normative Requirements			
The value shall be defined according to EBU Tech 3295 (e.g. SMPTE ST 12-1 time code).			

S0 2.9.10 StopProcessByServiceDefinedTimeType

complex type (base)		
Description		
A service-specific time mechanism specifies when a process is required to stop.		
Base	**Service Description**	**Content of Service Description**
bms:StopProcessType		
Normative Requirements		
The service shall wait for an event to stop the process. The actual event depends on the service implementation. A service shall support this type.		
Class Diagram		



actualStopTime

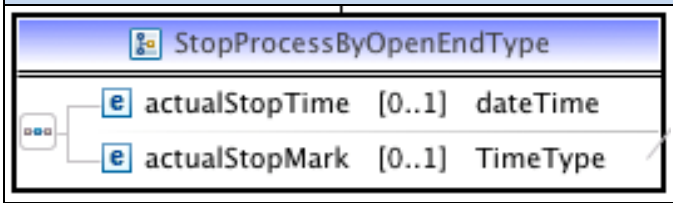
element (type)			
Description			
Actual time that the process stopped.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
If supported, the value shall be provided on completion of the process.			

actualStopMark

element (type)			
Description			
Time mark indicating the time that the process stopped as a value measured along the associated audio and/or video stream.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:TimeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

S0 2.9.11 StopProcessByOpenEndType

complex type (base)		
Description		
The real-time process is to continue indefinitely until a stop command (manageJobRequest) is received.		
Base	**Service Description**	**Content of Service Description**
bms:StopProcessType		
Normative Requirements		
Class Diagram		



actualStopTime

element (type)			
Description			
Actual time that the process stopped.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			
If supported, the value shall be provided on completion of the process.			

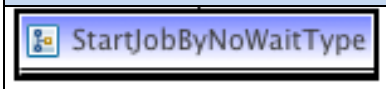
actualStopMark

element (type)			
Description			
Time mark indicating the time that the process stopped as a value measured along the associated audio and/or video stream.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:TimeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

S0 2.9.12 StartJobType

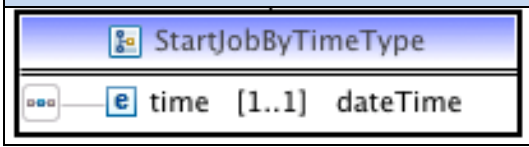
complex type (abstract)	
Description	
Kinds of time when a job should start.	
Service Description	**Content of Service Description**
Required	Supported start job types
Normative Requirements	
One of the following types shall be used. StartJobByNoWaitType shall be supported.	
Class Diagram	
	

S0 2.9.13 StartJobByNoWaitType

complex type (base)		
Description		
Start a job as soon as possible.		
Base	**Service Description**	**Content of Service Description**
bms:StartJobType		
Normative Requirements		
The job shall start without waiting for a specific time or event. This type shall be supported.		
Class Diagram		
		

S0 2.9.14 StartJobByTimeType

complex type (base)		
Description		
Provides a time when the job is required to start.		
Base	**Service Description**	**Content of Service Description**
bms:StartJobType		
Normative Requirements		
Class Diagram		



time

element (type)			
Description			
Time when the job is required start.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	dateTime	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.9.15 StartJobByLatestType

complex type (base)		
Description		
Start the job as close as possible to the defined start process time.		
Base	**Service Description**	**Content of Service Description**
bms:StartJobType		
Normative Requirements		
Class Diagram		
		

S0 2.10 FIMS-defined messages and faults

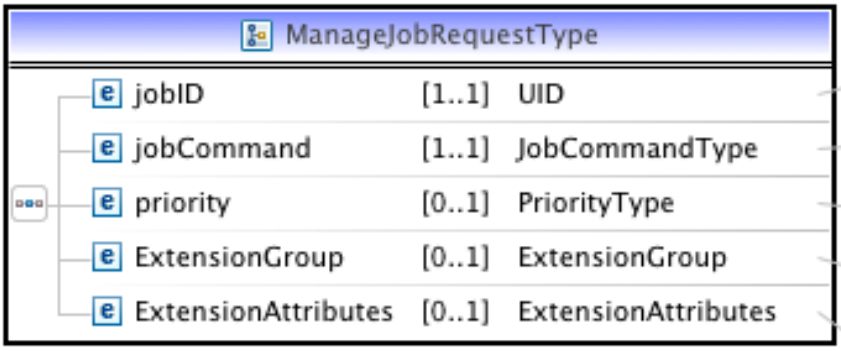
S0 2.10.1 ManageJobRequestType

complex type
Description
FIMS service common mechanism to manage jobs and update jobs properties, including job priority. The manage job request uses the jobCommand to set the intent of the command (acting on the job state).

Service Description	Content of Service Description

Normative Requirements

Class Diagram



jobID

element (type)			
Description			
Resource identifier of the job to be managed.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:UID	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

jobCommand

element (type)			
Description			
Action to perform on the job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:JobCommandType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

priority

element (type)			
Description			
New priority for the job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:PriorityType	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			
If the job command is set to "modifyPriority", this element shall be set to provide one of the allowed values (priority or profile). If the job command is set to any other value, this element shall not be set.			

ExtensionGroup

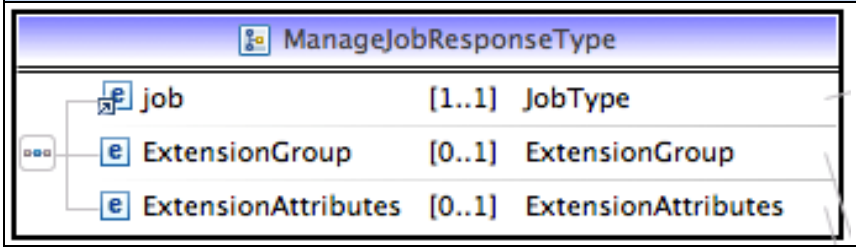
element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.2 ManageJobResponseType

complex type	
Description	
This type provides job status as a response to a job management request (e.g. pausing a job).	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



bms:job

element (ref)			
Description			
Job status as a response to a job management request.			
Occurrence	**Ref**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:job	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionGroup

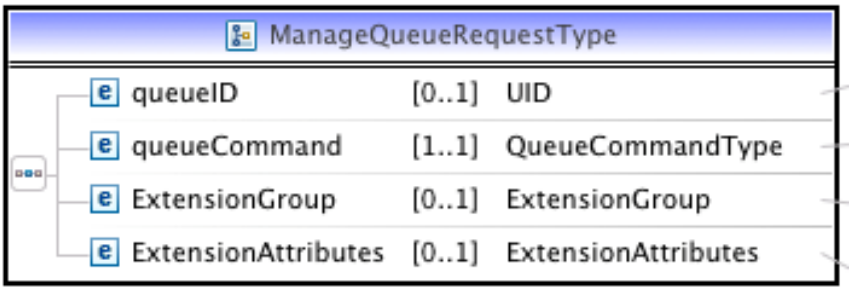
element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.3 ManageQueueRequestType

complex type	
Description	
This type provides a mechanism to manage a queue, for example checking the status of the queue, stopping and starting the queue and locking the queue. The queue management request uses the queueCommand to set the intent of the command (requesting status or acting on the queue state).	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



queueID

element (type)			
Description			
Identifier for the queue, required for services with more than one queue. Note: FIMS 1.0 does not allow more than one queue per service.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:UID	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

queueCommand

element (type)			
Description			
This element specifies the intent of the command. Possible values for the command are listed in QueueCommandType.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:QueueCommandType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionGroup

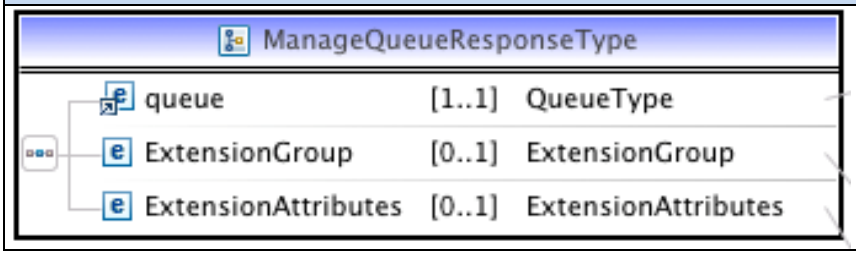
element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.4 ManageQueueResponseType

complex type	
Description	
This type represents a response to a queue management request. It provides information such as the status and the length of queue.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



bms:queue

element (ref)			
Description			
Queue information such as the status and the length of queue.			
Occurrence	**Ref**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:queue	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionGroup

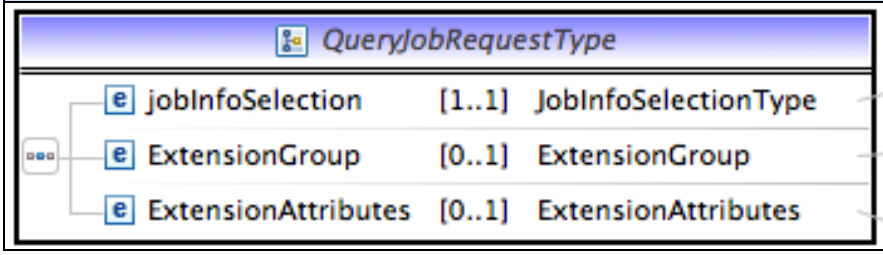
element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.5 QueryJobRequestType

complex type (abstract)	
Description	
This is an abstract type to be extended to create types to request status of multiple identified jobs or to list the status of jobs that meet requirements specified by a filter object.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



jobInfoSelection

element (type)			
Description			
Level of status details to be returned for the jobs.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:JobInfoSelectionType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

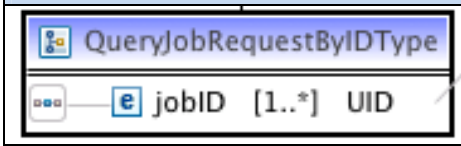
ExtensionGroup

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.6 QueryJobRequestByIDType

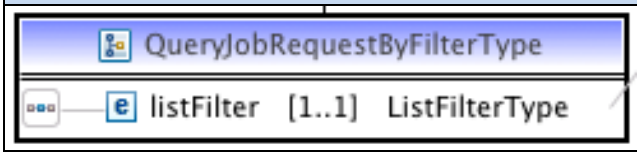
complex type (base)		
Description		
This type is part of the FIMS service common status interface and provides a mechanism query jobs and retrieve information associated to them. A client can request the status of multiple identified jobs.		
Base	**Service Description**	**Content of Service Description**
bms:QueryJobRequestType		
Normative Requirements		
Class Diagram		
		

jobID

element (type)			
Description			
This element carries the IDs of one or more jobs for which status is to be retrieved.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	bms:UID	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.7 QueryJobRequestByFilterType

complex type (base)		
Description		
This type is part of the FIMS service common status interface and provides a mechanism query jobs and retrieve information associated to them. A client can request the status of jobs that meet requirements specified by a filter object.		
Base	**Service Description**	**Content of Service Description**
bms:QueryJobRequestType		
Normative Requirements		
Class Diagram		



listFilter

element (type)			
Description			
This element defines the filtering parameters to list jobs for which status is to be retrieved.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:ListFilterType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

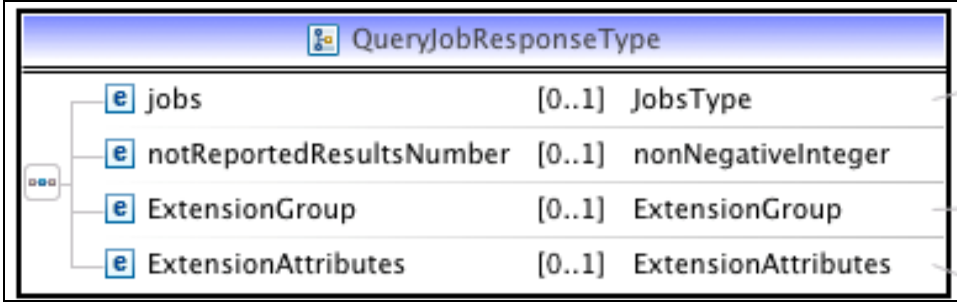
S0 2.10.8 QueryJobResponseType

complex type
Description
This type provides a response to QueryJobRequestType. It contains a list of queryJobInfo elements. Furthermore, the response message provided by this type supports a polling interaction pattern for retrieving the status and the response of an asynchronous job request.

Service Description	Content of Service Description

Normative Requirements

Class Diagram



jobs

element (type)			
Description			
List of jobs matching the query parameters.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:JobsType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

notReportedResultsNumber

element (type)			
Description			
This optional element indicates the number of results that were not reported due to the restriction imposed by the maximum number of results specified in the request.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionGroup

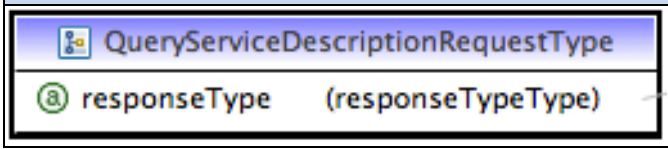
element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionGroup	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

ExtensionAttributes

element (type)			
Description			
Extension point.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:ExtensionAttributes	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.9 QueryServiceDescriptionRequestType

complex type	
Description	
Provides a mechanism to query Service Description.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	

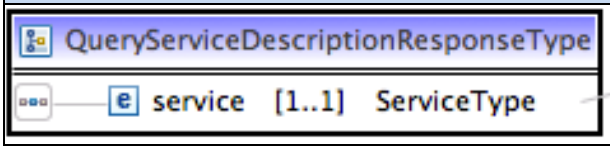


responseType

attribute (enum)			
Description			
Whether the response is required as an embedded description or as a reference.			
Use	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
Optional		Optional	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			
Default value shall be value.			
Enum Value	**Enum Description**		
ref	Response is required as a reference to the Service Description.		
value	Response is required as an embedded value with the complete Service Description.		

S0 2.10.10 QueryServiceDescriptionResponseType

complex type	
Description	
Response to the Service Description query.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



service

choice element (type)			
Description			
Service Description, embedded or by reference.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	ServiceType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.11 JobInfoSelectionType

simple type (enum)	
Description	
Whether to return only mandatory elements or all elements.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
mandatory	
all	

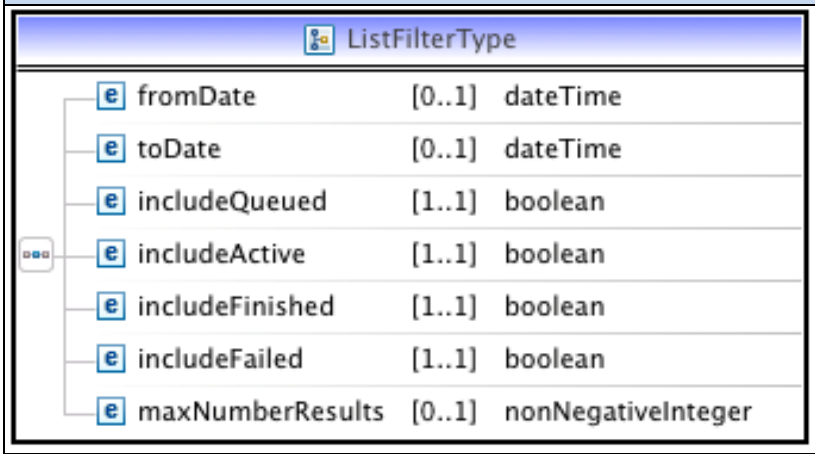
S0 2.10.12 ListFilterType

complex type
Description
Specifies a list of jobs to be retrieved. The list includes only jobs that fall in the period of time specified (if provided) and in the specified category (queued, active, completed, failed).

Service Description	Content of Service Description

Normative Requirements

Class Diagram



fromDate

choice element (type)			
Description			
A 'from' date filter. If this element is not specified, the list includes jobs that started at any point in time (that still complies with the 'toDate' field).			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			
Jobs to be listed shall have started on or after the date specified in this field.			

toDate

choice element (type)			
Description			
A 'to' date filter. If not provided, the list includes jobs that started at any point in time (that still complies with the 'fromDate' fileld).			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	dateTime	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			
Jobs to be listed shall have started on or before the date specified in this field.			

includeQueued

choice element (type)			
Description			
A flag to indicate job or jobs in the queue.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	boolean	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

includeActive

choice element (type)			
Description			
A flag to indicate job or jobs in the 'Running', 'Paused' or 'Unknown' state.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	boolean	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

includeFinished

choice element (type)			
Description			
A flag to indicate job or jobs in the 'Completed', 'Stopped' or 'Cleaned' state.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	boolean	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

includeFailed

choice element (type)			
Description			
A flag to indicate job or jobs in the 'Failed' state.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	boolean	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

maxNumberResults

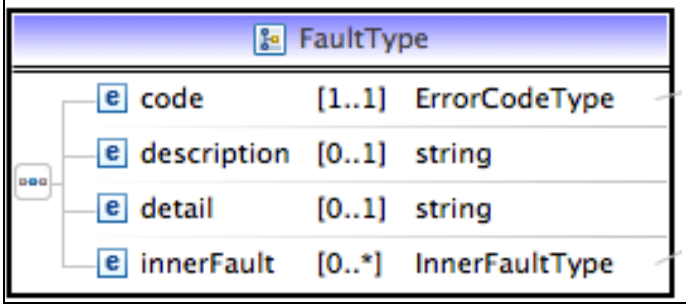
choice element (type)			
Description			
Maximum number of results to be listed.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	nonNegativeInteger	Optional	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.13 FaultType

complex type
Description
Details of a fault. This type can be extended by each service to provide additional error codes.

Service Description	Content of Service Description

Normative Requirements

Class Diagram


code

element (type)			
Description			
The error code specified in ErrorCodeType.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:ErrorCodeType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			
In the case of service-specific errors, "EXT_S00_0000" shall be set, and the specific error code shall be set in the child class "extendedCode" field.			

description

element (type)			
Description			
An optional description of the error.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

detail

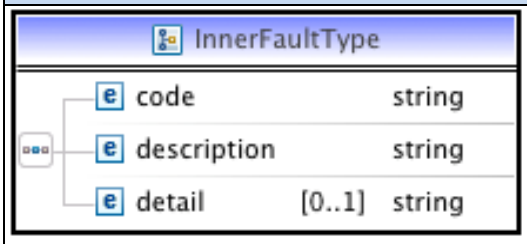
element (type)			
Description			
This optional field can provide a detailed description of the error.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

innerFault

element (type)			
Description			
Additional description of lower-level errors.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..*	bms:InnerFaultType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 2.10.14 InnerFaultType

complex type	
Description	
Additional description of lower-level errors.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	



code

element (type)			
Description			
Lower-level error code. This can be any value that is relevant to the appropriate subsystem			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	string	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

description

element (type)			
Description			
Description of lower-level error code. This can be any value that is relevant to the appropriate subsystem			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

detail

element (type)			
Description			
This optional field can provide a detailed description of the lower-level error.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

S0 3. EditUnitNumberType (Normative)

S0 3.1 Usage of combinations of editRate, factorNumerator, factorDenominator

For each of the actual rates listed in Tables 2 to 6, the corresponding combinations of editRate, factorNumerator and factorDenominator shall be as listed in each Table.

Table 2: Permitted combinations of editRate, factorNumerator, factorDenominator (Case: actual rate same as Nominal rate)

Nominal rate	Track	Actual rate (same as Nominal rate)		EditUnitNumberType		
	Audio (A) Video (V)	Rational	Rounded	editRate	factorNumerator	factorDenominator
24	A or V	24/1	24.000	24	1	1
25	A or V	25/1	25.000	25	1	1
30	A or V	30/1	30.000	30	1	1
48	A or V	48/1	48.000	48	1	1
50	A or V	50/1	50.000	50	1	1
60	A or V	60/1	60.000	60	1	1
44100	A	44100/1	44100	44100	1	1
48000	A	48000/1	48000	48000	1	1
88200	A	88200/1	88200	88200	1	1
96000	A	96000/1	96000	96000	1	1
176400	A	176400/1	176400	176000	1	1
192000	A	192000/1	192000	192000	1	1

Table 3: Permitted combinations of editRate, factorNumerator, factorDenominator (Case: 59.94i Pull Down)

Nominal rate	Track	Actual rate (59.94i Pull Down)		EditUnitNumberType		
	Audio (A) Video (V)	Rational	Rounded	editRate	factorNumerator	factorDenominator
24	A or V	24000/1001	23.976	24	1000	1001
25	A or V	25000/1001	24.975	25	1000	1001
30	A or V	30000/1001	29.970	30	1000	1001
48	A or V	48000/1001	47.952	48	1000	1001
50	A or V	50000/1001	49.950	50	1000	1001
60	A or V	60000/1001	59.940	60	1000	1001
44100	A	44100000/1001	44056	44100	1000	1001
48000	A	48000000/1001	47952	48000	1000	1001
88200	A	88200000/1001	88112	88200	1000	1001
96000	A	96000000/1001	95904	96000	1000	1001
176400	A	176400000/1001	176224	176000	1000	1001
192000	A	192000000/1001	191808	192000	1000	1001

Table 4: Permitted combinations of editRate, factorNumerator, factorDenominator (Case: 59.94i Pull up)

Nominal rate	Track	Actual rate (59.94i Pull Up)		EditUnitNumberType		
	Audio (A) Video (V)	Rational	Rounded	editRate	factorNumerator	factorDenominator
24	A or V	24024/1000	24.024	24	1001	1000
25	A or V	25025/1000	25.025	25	1001	1000
30	A or V	30030/1000	30.030	30	1001	1000
48	A or V	48048/1000	48.048	48	1001	1000
50	A or V	50050/1000	50.050	50	1001	1000
60	A or V	60060/1000	60.060	60	1001	1000
44100	A	44144100/1000	44144	44100	1001	1000
48000	A	48048000/1000	48048	48000	1001	1000
88200	A	88288200/1000	88288	88200	1001	1000
96000	A	96096000/1000	96096	96000	1001	1000
176400	A	176576400/1000	176576	176000	1001	1000
192000	A	192192000/1000	192192	192000	1001	1000

Table 5: Permitted combinations of editRate, factorNumerator, factorDenominator (Case: 50i Pull Down)

Nominal rate	Track	Actual rate (50i Pull Down)		EditUnitNumberType		
	Audio (A) Video (V)	Rational	Rounded	editRate	factorNumerator	factorDenominator
24	A or V	576/25	23.040	24	24	25
25	A or V	600/25	24.000	25	24	25
30	A or V	720/25	28.800	30	24	25
48	A or V	1152/25	46.080	48	24	25
50	A or V	1200/25	48.000	50	24	25
60	A or V	1440/25	57.600	60	24	25
44100	A	1058400/25	42336	44100	24	25
48000	A	1152000/25	46080	48000	24	25
88200	A	2116800/25	84672	88200	24	25
96000	A	2304000/25	92160	96000	24	25
176400	A	4233600/25	169344	176000	24	25
192000	A	4608000/25	184320	192000	24	25

Table 6: Permitted combinations of editRate, factorNumerator, factorDenominator (Case: 50i Pull Up)

Nominal rate	Track	Actual rate (50i Pull Up)		EditUnitNumberType		
	Audio (A) Video (V)	Rational	Rounded	editRate	factorNumerator	factorDenominator
24	A or V	600/24	25.000	24	25	24
25	A or V	625/24	26.042	25	25	24
30	A or V	750/24	31.250	30	25	24
48	A or V	1200/24	50.000	48	25	24
50	A or V	1250/24	52.083	50	25	24
60	A or V	1500/24	62.500	60	25	24
44100	A	1102500/24	45938	44100	25	24
48000	A	1200000/24	50000	48000	25	24
88200	A	2205000/24	91875	88200	25	24
96000	A	2400000/24	100000	96000	25	24
176400	A	4410000/24	183750	176000	25	24
192000	A	4800000/24	200000	192000	25	24

EBU

OPERATING EUROVISION

TECH 3356

SPECIFICATION OF THE FIMS MEDIA SOA FRAMEWORK

Part 2: Service Interfaces
S1, Transfer Service

VERSION 1.0.7

Published Jointly With FIMS and AMWA

Geneva
September 2012

[*] Page intentionally left blank. This document is paginated for two sided printing

Executive Summary

This document describes a vendor-neutral common framework for implementing Interoperable Media Services using a Service Oriented Architecture (SOA) based system, supporting interoperability, interchangeability and reusability of media specific services.

The FIMS 1.0 release at the time of publication of this document comprises the following:

- FIMS 1.0 Part 1: General Description:
  Part 01-General Description-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S0: Base Schema:
  Part 02_S0-Base Schema-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S1: Transfer Service (this document):
  Part 02_S1-Transfer Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S2: Transform Service:
  Part 02_S2-Transform Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S3: Capture Service:
  Part 02_S3-Capture Service-FIMS Schema Spec-1.0.7.pdf
- Schema file package:
  FIMS_1_0_7.zip

NOTES – The user's attention is called to the possibility that implementation and compliance with this specification may require use of subject matter covered by patent rights. By publication of this specification, no position is taken with respect to the existence or validity of any claim or of any patent rights in connection therewith. The AMWA, including the AMWA Board of Directors, shall not be responsible for identifying patents for which a license may be required by an AMWA specification or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for two sided printing

Framework for Interoperable Media Services
FIMS Media SOA Framework 1.0

Part 2, Supplement 1: Transfer Service

S1 1. Transfer Service Interface

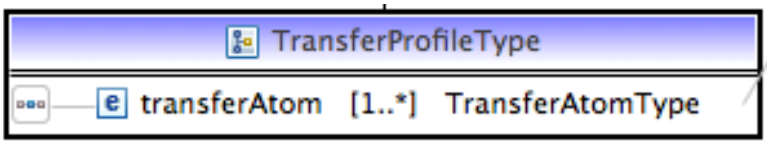
The Transfer Service Interface copies one or more files to another place (or to several places).

This document permits five different transfer protocols, HTTP, HTTPS, FTP, SFTP and FILE. A Transfer Service may implement one or more of these protocols, but a Transfer Service shall implement at least one protocol for an input and an output (although these do not need to match).

S1 1.1 FIMS-defined resources

S1 1.1.1 TransferProfileType

complex type (base)		
Description		
Profile used by the transfer media service to transfer files. The profile provides a mechanism to specify service-specific information for each service. To specify multiple destinations, multiple profiles are used through the profiles element of the transfer job.		
Base	**Service Description**	**Content of Service Description**
bms:ProfileType		
Normative Requirements		
Class Diagram		

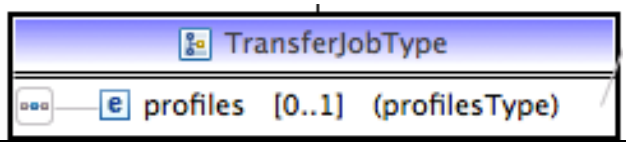


transferAtom

element (type)			
Description			
A set of parameters specific to the transfer media service, providing the transfer target destination.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	bms:TransferAtomType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S1 1.1.2 TransferJobType

complex type (base)		
Description		
Describes a transfer job.		
Base	**Service Description**	**Content of Service Description**
bms:JobType		
Normative Requirements		
Class Diagram		



profiles

element (child)			
Description			
Transfer profiles associated with this job.			
Occurrence	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	transferProfile	Mandatory	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

transferProfile

element (type)			
Description			
Transfer profiles associated with this job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	tms:TransferProfileType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S1 1.2 FIMS-defined enumerations

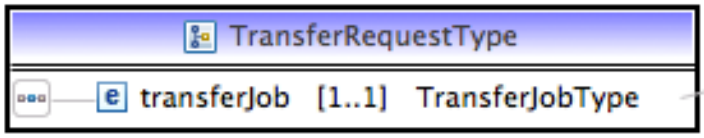
S1 1.2.1 TransferErrorCodeType

simple type (enum)	
Description	
Specific error codes for the transfer service: • INF_S01_xxxx: Infrastructure errors (system, storage, network, memory, processor) • DAT_S01_xxxx: Data errors (validation, missing, duplication) • SVC_S01_xxxx: Operation errors (existence, support, lock, connection, failure) • SEC_S01_xxxx: Security errors (authentication, authorization)	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
DAT_S01_0001	Invalid URI protocol specified for Transport operations.
DAT_S01_0002	Invalid OutpurDirectory or Target URI path.
DAT_S01_0003	Incorrect Hash. File received does not have same hash as specified in the file hash value property.
SVC_S01_0001	Unsupported protocol.
SVC_S01_0002	Unsupported hash type.
SVC_S01_0003	Encryption not supported.
SVC_S01_0004	Authentication not supported.
SVC_S01_0005	Integrity check not supported.
SVC_S01_0006	File too large.
SVC_S01_0007	Times not possible.
SVC_S01_0008	Incorrect file size.
SVC_S01_0009	Rejected by operator.
SVC_S01_0010	Transfer process ended unexpectedly.
INF_S01_0001	Network link with insufficient bandwidth.
INF_S01_0002	Link timed out.

S1 1.3 FIMS-defined messages and faults

S1 1.3.1 TransferRequestType

complex type	
Description	
A request to the transfer media service to transfer content from a source location to a target location.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	

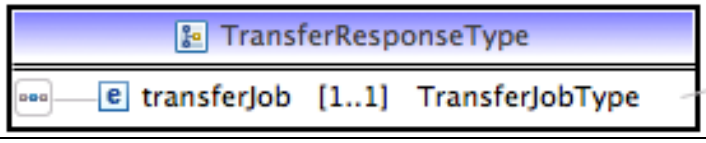


transferJob

element (type)			
Description			
Describes a transfer job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tms:TransferJobType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S1 1.3.2 TransferResponseType

complex type	
Description	
An acknowledgment sent back from the transfer media service when the transfer job is submitted.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	

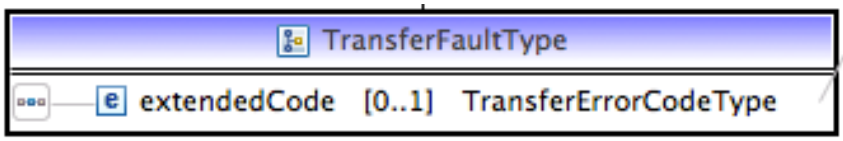


transferJob

element (type)			
Description			
Describes a transfer job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tms:TransferJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S1 1.3.3 TransferFaultType

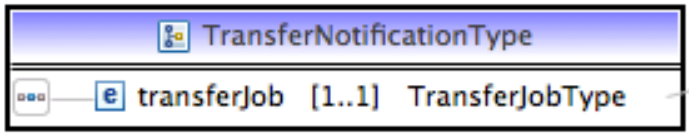
complex type (base)		
Description		
Fault information for the transfer media service.		
Base	**Service Description**	**Content of Service Description**
bms:FaultType		
Normative Requirements		
If an exception is generated when the transfer request message is submitted to the service, it shall respond with a message based on the TransferFaultType.		
Class Diagram		



extendedCode

element (type)			
Description			
Extended code for the transfer fault.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	tms:TransferErrorCodeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			
This element shall be set if the service needs to provide service-specific error codes which are not specified in the fault 'code' element. In this case, the code element shall be set to 'EXT_S00_0000' and the extendedCode shall be set to one of the codes described in TransferErrorCodeType. If the 'code' element is set to any other value than 'EXT_S00_0000' then this element shall not be set.			

S1 1.3.4 TransferNotificationType

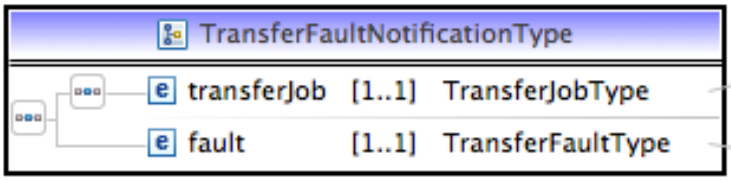
complex type	
Description	
Notification of the successful completion of a transfer job.	
Service Description	**Content of Service Description**
Normative Requirements	
If one or more 'notifyAt' elements are set for the associated transfer job, then the service shall respond with a notification to the specified endpoints when the job completes. (If notification is supported.)	
Class Diagram	
	

transferJob

element (type)			
Description			
Describes a transfer job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tms:TransferJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S1 1.3.5 TransferFaultNotificationType

complex type	
Description	
Fault notification for the transfer service, including the related transfer job and the fault code.	
Service Description	**Content of Service Description**
Normative Requirements	
If one or more “notifyAt” elements are set for the transfer job and a failure occurs during the job execution, then the service shall respond with a transfer fault notification to the endpoint specified by 'faultTo’.	
Class Diagram	



transferJob

element (type)			
Description			
Describes a transfer job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tms:TransferJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

fault

element (type)			
Description			
The fault element.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tms:TransferFaultType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

EBU

OPERATING EUROVISION

TECH 3356

SPECIFICATION OF THE FIMS MEDIA SOA FRAMEWORK

Part 2: Service Interfaces
S2, Transform Service

VERSION 1.0.7

Published Jointly With FIMS and AMWA

Geneva
September 2012

[*] Page intentionally left blank. This document is paginated for two sided printing

Executive Summary

This document describes a vendor-neutral common framework for implementing Interoperable Media Services using a Service Oriented Architecture (SOA) based system, supporting interoperability, interchangeability and reusability of media specific services.

The FIMS 1.0 release at the time of publication of this document comprises the following:

- FIMS 1.0 Part 1: General Description:
  Part 01-General Description-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S0: Base Schema:
  Part 02_S0-Base Schema-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S1: Transfer Service:
  Part 02_S1-Transfer Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S2: Transform Service (this document):
  Part 02_S2-Transform Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S3: Capture Service:
  Part 02_S3-Capture Service-FIMS Schema Spec-1.0.7.pdf
- Schema file package:
  FIMS_1_0_7.zip

NOTES – The user's attention is called to the possibility that implementation and compliance with this specification may require use of subject matter covered by patent rights. By publication of this specification, no position is taken with respect to the existence or validity of any claim or of any patent rights in connection therewith. The AMWA, including the AMWA Board of Directors, shall not be responsible for identifying patents for which a license may be required by an AMWA specification or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for two sided printing

Framework for Interoperable Media Services
FIMS Media SOA Framework 1.0

Part 2, Supplement 2: Transform Service

S2 1. Transform Service Interface

A transform Service Interface operates on media content to transform the content or a list of content elements from a source media format to a target or a list of target media formats. It builds on the Transfer Service interface, and adds some element of transformation to the input.

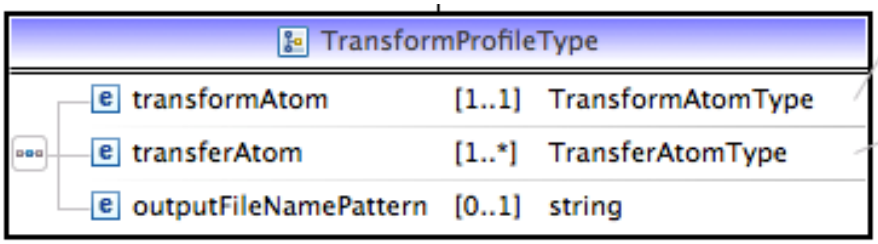
This document defines alteration of essence and container formats (including technical metadata). Specific AV Processes will be specified in a future version of the FIMS specification.

The Transform Service can process Transfer atoms embedded within a Transform profile. However, a Transform request not carrying out any Transform operations is just a Transfer request – and shall use a Transfer request instead.

S2 1.1 FIMS-defined resources

S2 1.1.1 TransformProfileType

complex type (base)		
Description		
Profile used by the transform media service to perform a transformation on media content. The profile specifies the media format to be output.		
Base	**Service Description**	**Content of Service Description**
bms:ProfileType		
Normative Requirements		
Class Diagram		



transformAtom

element (type)			
Description			
Set of parameters specific to the transform media service, describing the media format to be output.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:TransformAtomType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

transferAtom

element (type)			
Description			
Set of parameters specific to transfer aspects of the transform service, providing a destination for output files.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	bms:TransferAtomType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

outputFileNamePattern

element (type)			
Description			
Specifies a name for the file and, where applicable, a pattern for the names of the files to be output.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not for this parameter. When multiple output files are produced, include details of the patterns.		
Normative Requirements			

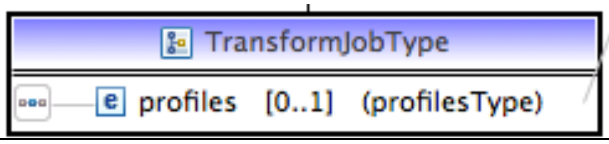
If the name is not provided, the service shall automatically generate a file name.

The pattern should be specified using a regular expression (IEEE POSIX Basic Regular Expressions (BRE) standard) to specify a pattern to build the output file name. The regular expression should allow to reference the following variables in the expression:

- Source file name;
- Date;
- Profile name used for the transformation;
- Format name used for the transformation;
- Frame reference;
- AutoIncrementCounter that is incremented by 1 for each new file.

S2 1.1.2 TransformJobType

complex type (base)		
Description		
Describes a transform job.		
Base	**Service Description**	**Content of Service Description**
bms:JobType		
Normative Requirements		
Class Diagram		



profiles

element (child)			
Description			
Transform profiles associated with this job.			
Occurrence	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	transformProfile	Mandatory	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

transformProfile

element (type)			
Description			
Transform profiles associated with this job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	tfms:TransformProfileType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

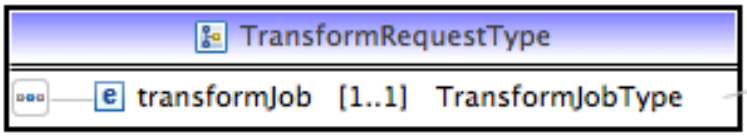
S2 1.2 FIMS-defined enumerations

S2 1.2.1 TransformErrorCodeType

simple type (enum)	
Description	
Specific error codes for the transform service • INF_S02_xxxx: Infrastructure errors (system, storage, network, memory, processor) • DAT_S02_xxxx: Data errors (validation, missing, duplication) • SVC_S02_xxxx: Operation errors (existence, support, lock, connection, failure) • SEC_S02_xxxx: Security errors (authentication, authorization)	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
SVC_S02_0001	Invalid target media format.

S2 1.3 FIMS-defined messages and faults

S2 1.3.1 TransformRequestType

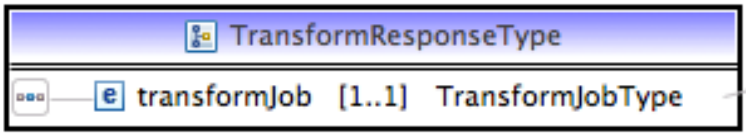
complex type	
Description	
A request to the transform media service to transform content from a source format to a target format.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	
	

transformJob

element (type)			
Description			
Describes a transform job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tfms:TransformJobType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S2 1.3.2 TransformResponseType

complex type	
Description	
Represents the acknowledgment sent back from the transform media service when the transform job is submitted.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	

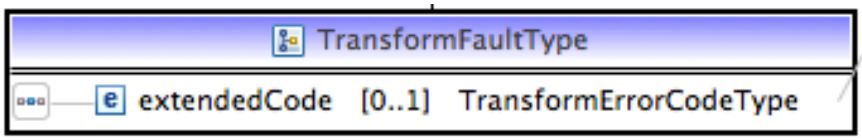


transformJob

element (type)			
Description			
Describes a transform job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tfms:TransformJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S2 1.3.3 TransformFaultType

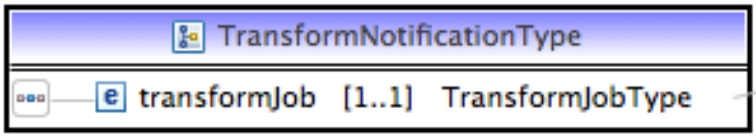
complex type (base)		
Description		
Fault information for the transform media service.		
Base	**Service Description**	**Content of Service Description**
bms:FaultType		
Normative Requirements		
If an exception is generated when the transform request message is submitted to the service, it shall respond with a message based on the TransformFaultType.		
Class Diagram		



extendedCode

element (type)			
Description			
Transform service-specific error codes, as detailed in the TransformErrorCodeType.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	tfms:TransformErrorCodeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			
This element shall be set if the service needs to provide service-specific error codes which are not specified in the fault 'code' element. In this case, the code element shall be set to 'EXT_S00_0000' and the extendedCode shall be set to one of the codes described in TransformErrorCodeType. If the 'code' element is set to any other value than 'EXT_S00_0000' then this element shall not be set.			

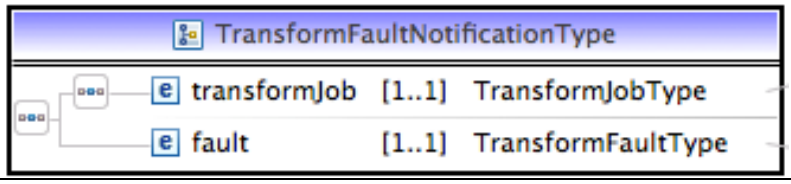
S2 1.3.4 TransformNotificationType

complex type	
Description	
Notification of the successful completion of a transform job.	
Service Description	**Content of Service Description**
Normative Requirements	
If one or more 'notifyAt' elements are set for the associated transform job, then the service shall respond with a notification to the specified endpoints when the job completes. (If notification is supported.)	
Class Diagram	
	

transformJob

element (type)			
Description			
Describes a transform job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tfms:TransformJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S2 1.3.5 TransformFaultNotificationType

complex type	
Description	
Fault notification for the transform service, including the related transform job and the fault code.	
Service Description	
Normative Requirements	
If one or more “notifyAt” elements are set for the transform job and a failure occurs during the job execution, then the service shall respond with a transform fault notification to the endpoint specified by 'faultTo’.	
Class Diagram	
	

transformJob

element (type)			
Description			
Describes a transform job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tfms:TransformJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

fault

element (type)			
Description			
Specific fault information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	tfms:TransformFaultType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

EBU

OPERATING EUROVISION

TECH 3356

SPECIFICATION OF THE FIMS MEDIA SOA FRAMEWORK

Part 2: Service Interfaces
S3, Capture Service

VERSION 1.0.7

Published Jointly With FIMS and AMWA

Geneva
September 2012

[*] Page intentionally left blank. This document is paginated for two sided printing

Executive Summary

This document describes a vendor-neutral common framework for implementing Interoperable Media Services using a Service Oriented Architecture (SOA) based system, supporting interoperability, interchangeability and reusability of media specific services.

The FIMS 1.0 release at the time of publication of this document comprises the following:

- FIMS 1.0 Part 1: General Description:
  Part 01-General Description-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S0: Base Schema:
  Part 02_S0-Base Schema-FIMS Schema Spec-1.0.7-Rev1.pdf
- FIMS 1.0 Part 2, S1: Transfer Service:
  Part 02_S1-Transfer Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S2: Transform Service:
  Part 02_S2-Transform Service-FIMS Schema Spec-1.0.7.pdf
- FIMS 1.0 Part 2, S3: Capture Service (this document):
  Part 02_S3-Capture Service-FIMS Schema Spec-1.0.7.pdf
- Schema file package:
  FIMS_1_0_7.zip

NOTES – The user's attention is called to the possibility that implementation and compliance with this specification may require use of subject matter covered by patent rights. By publication of this specification, no position is taken with respect to the existence or validity of any claim or of any patent rights in connection therewith. The AMWA, including the AMWA Board of Directors, shall not be responsible for identifying patents for which a license may be required by an AMWA specification or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for two sided printing

Framework for Interoperable Media Services
FIMS Media SOA Framework 1.0

Part 2, Supplement 3: Capture Service

S3 1. Capture Service Interface

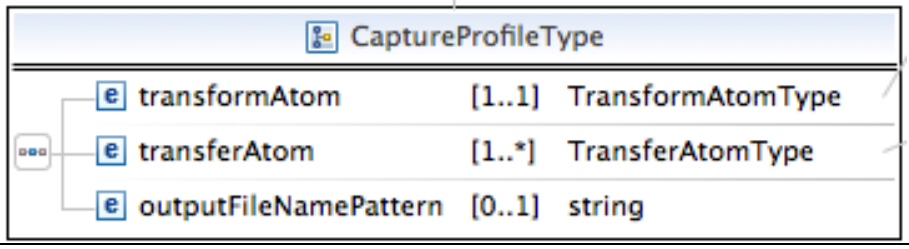
The Capture Service Interface creates one or more related content items from video and audio sources. The service may involve a human operator to start and stop the operation. It builds on the Transform Service interface, but is constrained to support isochronous stream-based input, for example an HD-SDI stream.

In the case of audiovisual signals, the service can simply encode and wrap these into a file or can encode and wrap them with multiple formats into multiple files and put them in several places. The service could even perform an AV Process such as resize, burn-in, and LUT before encoding. In a sense, the Capture Service by nature includes elements of both a Transform Service and a Transfer Service within it.

S3 1.1 FIMS-defined resources

S3 1.1.1 CaptureProfileType

complex type (base)		
Description		
Profile used by a capture media service to capture media content. The profile specifies how to make the capture, what media format(s) to be output, and how it is processed.		
Base	**Service Description**	**Content of Service Description**
bms:ProfileType		
Normative Requirements		
Class Diagram		



transformAtom

element (type)			
Description			
Set of parameters that describe the output media format for the captured clip.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:TransformAtomType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

transferAtom

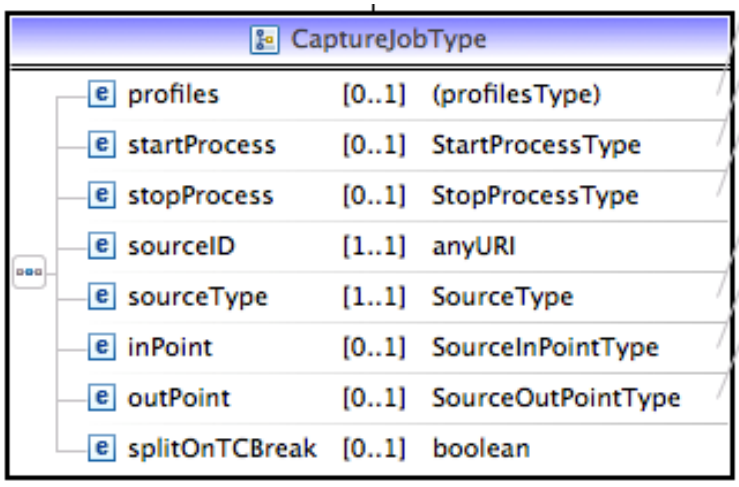
element (type)			
Description			
Set of transfer parameters for the capture service, used to specify the target location for the output.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	bms:TransferAtomType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

outputFileNamePattern

element (type)			
Description			
Specifies a name for the file and, where applicable, a pattern for the names of the files to be output.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	string	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not for this parameter. When multiple output files are produced, include details of the patterns.		
Normative Requirements			
If the name is not provided, the service shall automatically generate a file name. The pattern should be specified using a regular expression (IEEE POSIX Basic Regular Expressions (BRE) standard) to specify a pattern to build the output file name. The regular expression should allow to reference the following variables in the expression: • Profile name used for the transformation; • Format name used for the transformation; • Frame number; • An index or counter which is incremented by 1 for each new file.			

S3 1.1.2 CaptureJobType

complex type (base)		
Description		
Describes a capture job.		
Base	**Service Description**	**Content of Service Description**
bms:ProfileType		
Normative Requirements		
Class Diagram		



profiles

element (child)			
Description			
Capture profiles associated with this job.			
Occurrence	**Child**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	captureProfile	Mandatory	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			

captureProfile

element (type)			
Description			
Capture profiles associated with this job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..*	cms:CaptureProfileType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

startProcess

element (type)			
Description			
Specifies the system time at the start of the stream process. There are the following four types to specify the time parameter of startProcess, which could need to be frame accurate: • NoWait type: execute immediately; • Time type: the time to start; • TimeMark type: the time at which the TimeMark embedded in essence such as a timecode is detected; • ServiceDefinedTime type: the time defined by a service.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:StartProcessType	Mandatory	Optional
Service Description	**Contents of Service Description**		
Required	Kinds of start process supported.		
Normative Requirements			
A service which supports Time and/or TimeMark type shall declare this in the Service Description.			

stopProcess

element (type)			
Description			
Specifies the process stop time. There are the following five types to specify the time parameter of stopProcess, which could need to be frame accurate: • OpenEnd type: the time at which a process stop command (manageJobRequest) is received; • Time type: the time to stop; • TimeMark type: the time at which the TimeMark embedded in essence such as a timecode is detected; • Duration type: the time at which point the specified duration has elapsed since startProcess; • ServiceDefinedTime type: the time defined by a service.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	bms:StopProcessType	Mandatory	Optional
Service Description	**Contents of Service Description**		
Required	Kinds of stop process supported.		
Normative Requirements			

sourceID

element (type)			
Description			
The identification of the source.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	anyURI	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Required	Configuration description included in the Service Description.		
Normative Requirements			

sourceType

element (type)			
Description			
Type of the source, specified by the source type enumeration (e.g. "controllable" such as a VTR or "uncontrolled" such as a feed).			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	cms:SourceType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Required	Configuration description included in the Service Description.		
Normative Requirements			
If a service does not support the required feature type, it shall return either 'Feature not supported' error code (SVC_S00_0015) or 'Operation requested is not currently supported by the service or the device' error code (SVC_S00_0003).			

inPoint

element (type)			
Description			
"In" point of a controllable capture source, such as a VTR.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	cms:SourceInPointType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not. If supported, the kinds of SourceInPointType.		
Normative Requirements			

outPoint

element (type)			
Description			
"Out" point of a controllable capture source, such as a VTR.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	cms:SourceOutPointType	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not. If supported, the kinds of SourceOutPointType.		
Normative Requirements			

splitOnTCBreak

element (type)			
Description			
Whether the output is split into multiple essence files whenever there is a timecode discontinuity. Use the value "true" to specify that whenever a timecode discontinuity is found, the output is split into multiple essence files. Use the value "false" to specify that even when a timecode discontinuity is found, the output is not split but is created as a single essence file.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	boolean	Optional	Optional
Service Description	**Contents of Service Description**		
Required	Support or not.		
Normative Requirements			

S3 1.2 FIMS-defined enumerations

S3 1.2.1 SourceType

simple type (enum)	
Description	
Whether a connected source is controllable or uncontrollable.	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
controllable	A controllable source, such as a VTR.
uncontrolled	An uncontrollable source, such as a feed.

S3 1.2.2 CaptureErrorCodeType

simple type (enum)	
Description	
Specific error codes for the capture service • INF_S03_xxxx: Infrastructure errors (system, storage, network, memory, processor) • DAT_S03_xxxx: Data errors (validation, missing, duplication) • SVC_S03_xxxx: Operation errors (existence, support, lock, connection, failure) • SEC_S03_xxxx: Security errors (authentication, authorization)	
Service Description	**Content of Service Description**
Normative Requirements	
Enum Value	**Enum Description**
SVC_S03_0001	Invalid target media format.
SVC_S03_0002	Inconsistent time constraints.
DAT_S03_0001	Invalid source ID.

S3 1.3 FIMS-defined control data types

S3 1.3.1 SourceInPointType

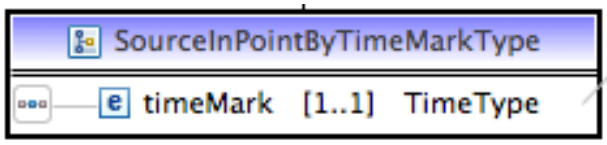
complex type (abstract)	
Description	
Kinds of source “In” point used as a parameter for a capture process.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	
	

S3 1.3.2 SourceInPointByCurrentType

complex type (base)		
Description		
The current position of the capture source used as the source “In” point.		
Base	**Service Description**	**Content of Service Description**
cms:SourceInPointType		
Normative Requirements		
Class Diagram		
		

S3 1.3.3 SourceInPointByTimeMarkType

complex type (base)		
Description		
Source “In” point specified by a time mark in the capture source essence.		
Base	**Service Description**	**Content of Service Description**
cms:SourceInPointType		
Normative Requirements		
Class Diagram		



timeMark

element (type)			
Description			
Source “In” point specified by a timecode or similar time mark.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:TimeType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S3 1.3.4 SourceInPointByBeginningType

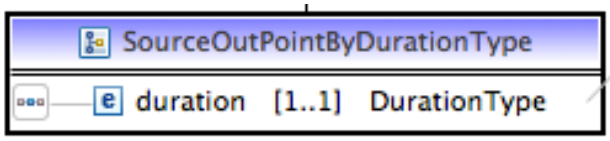
complex type (base)		
Description		
Source “In” point is at the beginning of the capture source, such as VTR tape.		
Base	**Service Description**	**Content of Service Description**
cms: SourceInPointType		
Normative Requirements		
Class Diagram		
		

S3 1.3.5 SourceOutPointType

complex type (abstract)	
Description	
Kinds of source “Out” point used as a parameter for a capture process.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	
	

S3 1.3.6 SourceOutPointByDurationType

complex type (base)		
Description		
Source “Out” point specified by the total duration of material to capture.		
Base	**Service Description**	**Content of Service Description**
cms: SourceOutPointType		
Normative Requirements		
Class Diagram		

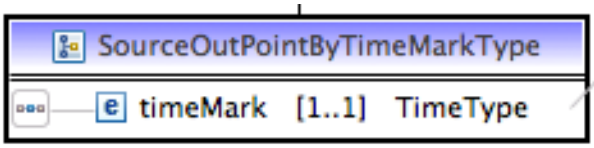


duration

element (type)			
Description			
Total duration of material to capture, specified by timecode, real time or number of edit units.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:DurationType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S3 1.3.7 SourceOutPointByTimeMarkType

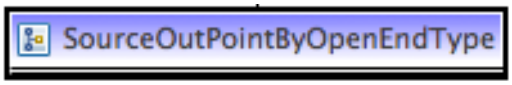
complex type (base)		
Description		
Source “Out” point specified by a position in the capture source essence.		
Base	**Service Description**	**Content of Service Description**
cms: SourceOutPointType		
Normative Requirements		
Class Diagram		



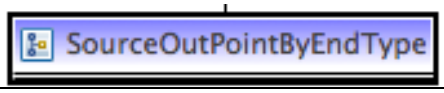
timeMark

element (type)			
Description			
Source “Out” point specified by a timecode or similar time mark.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	bms:TimeType	Mandatory	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S3 1.3.8 SourceOutPointByOpenEndType

complex type (base)		
Description		
Source "Out" point is not specified and capture continues indefinitely until a stop command (manageJobRequest) is received.		
Base	**Service Description**	**Content of Service Description**
cms: SourceOutPointType		
Normative Requirements		
Class Diagram		
		

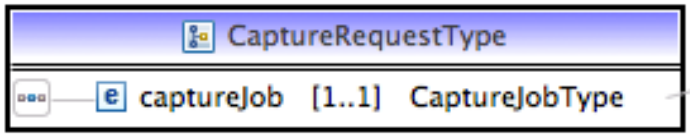
S3 1.3.9 SourceOutPointByEndType

complex type (base)		
Description		
Source “Out” point is the end of the capture source, such as VTR tape.		
Base	**Service Description**	**Content of Service Description**
cms: SourceOutPointType		
Normative Requirements		
Class Diagram		
		

S3 1.4 FIMS-defined messages and faults

S3 1.4.1 CaptureRequestType

complex type	
Description	
A request to the capture media service to transform content from a source format to a target format.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	

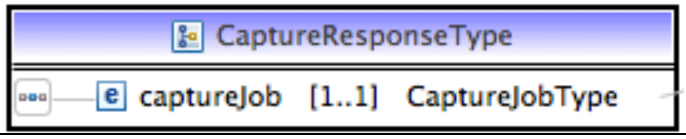


captureJob

element (type)			
Description			
Set of parameters that describe the output media format.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	cms:CaptureJobType	Mandatory	Not applicable
Service Description	**Contents of Service Description**		
Normative Requirements			

S3 1.4.2 CaptureResponseType

complex type	
Description	
A response to a capture media service request for content capture from a feed source to a target format.	
Service Description	**Content of Service Description**
Normative Requirements	
Class Diagram	

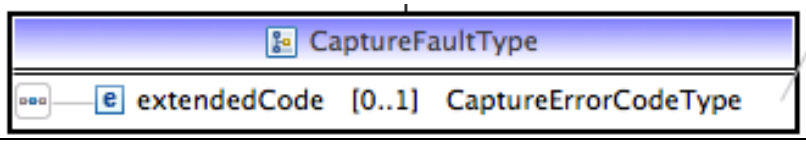


captureJob

element (type)			
Description			
A response to a capture media service request for content capture from a feed source to a target format.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	cms:CaptureJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S3 1.4.3 CaptureFaultType

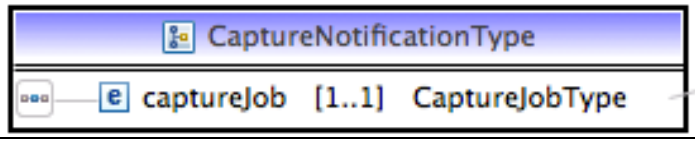
complex type (base)		
Description		
Fault information for the capture media service. It extends the base FaultType with a complementary extended code that allows service-specific error codes to be included if needed.		
Base	**Service Description**	**Content of Service Description**
bms:FaultType		
Normative Requirements		
If an exception is generated when the capture request message is submitted to the service, it shall respond with a message based on the CaptureFaultType.		
Class Diagram		



extendedCode

element (type)			
Description			
Extended error code for the capture fault.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
0..1	cms:CaptureErrorCodeType	Not applicable	Optional
Service Description	**Contents of Service Description**		
Normative Requirements			
This element shall be set if the service needs to provide service-specific error codes which are not specified in the fault 'code' element. In this case, the code element shall be set to 'EXT_S00_0000' and the extendedCode shall be set to one of the codes described in CaptureErrorCodeType. If the 'code' element is set to any other value than 'EXT_S00_0000' then this element shall not be set.			

S3 1.4.4 CaptureNotificationType

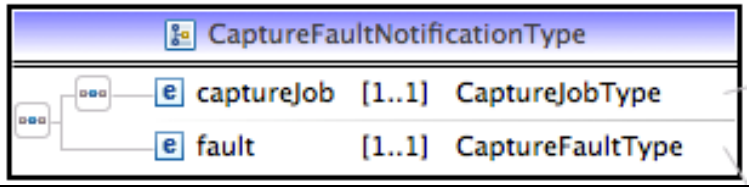
complex type	
Description	
Notification of the successful completion of a capture job.	
Service Description	**Content of Service Description**
Normative Requirements	
If one or more 'notifyAt' elements are set for the associated capture job, then the service shall respond with a notification to the specified endpoints when the job completes. (If notification is supported.)	
Class Diagram	
	

captureJob

element (type)			
Description			
Describes a capture job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	cms:CaptureJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

S3 1.4.5 CaptureFaultnotificationType

complex type	
Description	
Fault notification for the capture service, including the related capture job and the fault code.	
Service Description	**Content of Service Description**
Normative Requirements	
If one or more “notifyAt” elements are set for the capture job and a failure occurs during the job execution, then the service shall respond with a capture fault notification to the endpoint specified by 'faultTo’.	
Class Diagram	



captureJob

element (type)			
Description			
Describes a capture job.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	cms:CaptureJobType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			

fault

element (type)			
Description			
Fault information.			
Occurrence	**Type**	**Inclusion (Req.)**	**Inclusion (Res.)**
1..1	cms:CaptureFaultType	Not applicable	Mandatory
Service Description	**Contents of Service Description**		
Normative Requirements			