Asset Publishing Interface

Generic Asset Publishing

Version 1.0.0

Status: Proposed

Filename	AssetPublishing-Interface-v1.0.0.docx
Date	14 Sept 2016
Author(s)	Saithala RaviSankar Kalyan
Information Domain	Content Processing Workflows
Client/Project	ML 2.x
Owner	MSPU Nagravision

CH-1033 Cheseaux, Switzerland.
Tel: +41 21 7320311 Fax: +41 21 7320100
www.nagra.com

This document is supplied with an understanding that the notice(s) herein or any other contractual agreement(s) made that instigated the delivery of a hard copy, electronic copy, facsimile or file transfer of this document are strictly observed and maintained.

The information contained in this document is subject to change without notice.

Security Policy of Nagravision Kudelski Group

Any recipient of this document, without exception, is subject to a Non-Disclosure Agreement (NDA) and access authorization.

Contents

List of figures

Preface

Document purpose

This document describes the basis of the Asset Publishing interface that will be used to drive asset publish and purge activities.

Document change policy

It is defined that the versioning of this document is based on three parts incremented as follow:

- The first part is incremented when a major or non-backward compatible evolution of the interface occurs.
- The second part is incremented when a backward compatible evolution of the interface occurs. For example addition of a new optional parameter in an existing method.
- The third part is incremented when the changes impact only the documentation (for instance a correction in an explanation or in the typography)

Used Conventions

When the document references an XML element within the text, this element is written in another font and is represented between angle brackets (e.g. <Content>).

Other references to the same word make refer to the Media Live CMS interpretation of this word, for this example, where the word is written in normal text font "Content" describes all the editorial and eventually technical information of a movie itself.

Following the same idea, the attribute name is also written in another font.

Document History

Change logs
0.0.1 / 2015.06.26 / Sameera Nimmala, RaviSankar Saithala ▪ Draft version **0.0.2 / 2015.07.06 / Sameera Nimmala, RaviSankar Saithala** ▪ Incorporated first review comments **1.0.0 / 2015.07.29 / Sameera Nimmala, RaviSankar Saithala** ▪ First proposed version

Document Reviewers

Name	Function	Review Date
Sanchez Christophe	Software Architect	
Molyneaux Wendy	System Architect	2015.07.06

Document Approvers

Name	Function	Approval Date
Ferreira Pedro	Product Line Manger	

1. Overview

1.1 API context

This document describes the API (Publishing Interface) between the Nagra MediaLive WFM and the third party who implements this interface. The Asset Publishing interface supports below basic functionalities:

- LoadAsset
    - Publishes the asset file from source to a destination CDN based on the URL provided. The asset file may be a single file (eg: *.ts) or a directory.
- PurgeAsset
    - Purges the asset file (eg: *.ts or directory) from the destination CDN.

Note

Third parties can implement either of the methods LoadAsset or PurgeAsset of the Publishing Interface or both the methods. If a third party is implementing only PurgeAsset method then the location of the asset that is published on CDN should be synchronized with asset file location that will be used by purge method.

1.2 Use cases

1.2.1 LoadAsset

For each CMS publish job found, the Workflow Manager constructs a request message, containing “SourceAssetURL”, “DestinationAsset” & “MetaData” and makes SOAP call to Generic Asset Publishing server. Publishing server returns a “TaskId” as part of response message. This “TaskId” will be used to retrieve the status of the LoadAsset task. Based on the publishing job status (i.e, SUCCESS/FAILED) WFM updates the CMS publish job status (i.e, DONE/FAILED) in CMS DB.

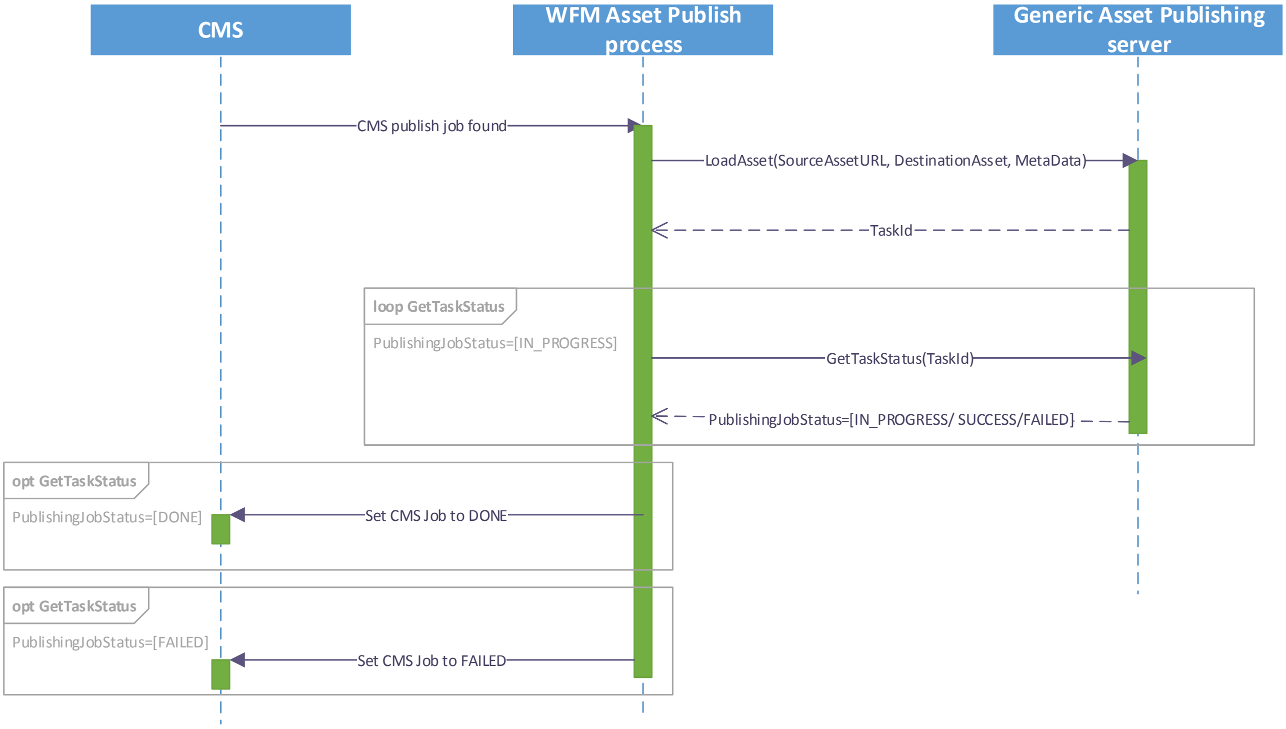

CMS
WFM Asset Publish process
Generic Asset Publishing server
CMS publish job found
LoadAsset(SourceAssetURL, DestinationAsset, MetaData)
TaskId
loop GetTaskStatus
PublishingJobStatus=[IN_PROGRESS]
GetTaskStatus(TaskId)
PublishingJobStatus=[IN_PROGRESS/ SUCCESS/FAILED]
opt GetTaskStatus
PublishingJobStatus=[DONE]
Set CMS Job to DONE
opt GetTaskStatus
PublishingJobStatus=[FAILED]
Set CMS Job to FAILED

Figure 1: Basic call sequence for LoadAsset

Load asset task state diagram

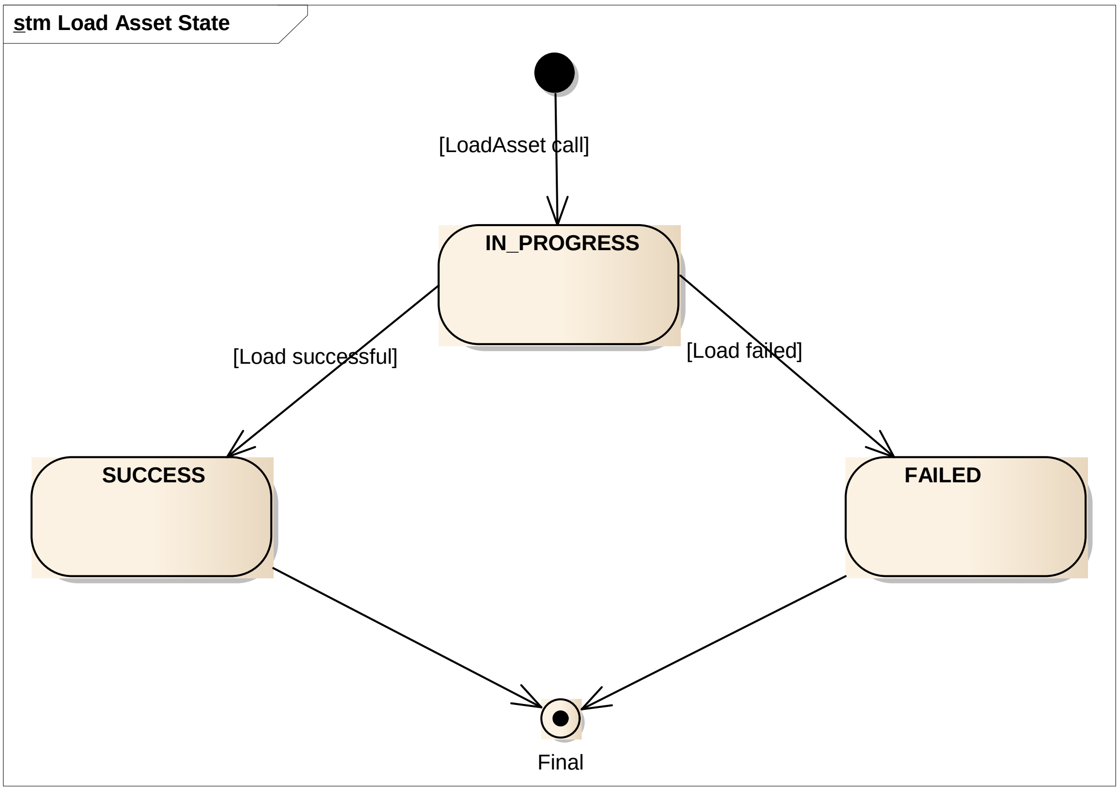


Figure 2: Load Asset State Diagram

1.2.2 PurgeAsset

For each CMS purge job found, the Workflow Manager constructs a request message, containing “DestinationAsset” & “MetaData”, and makes SOAP call to Generic Asset Publishing server. Publishing server returns a “TaskId” as part of response message. This “TaskId” will be used to retrieve the status of the PurgeAsset task. Based on the purge job status (i.e, SUCCESS/FAILED) WFM updates the CMS purge job status (i.e, DONE/FAILED) in CMS DB.

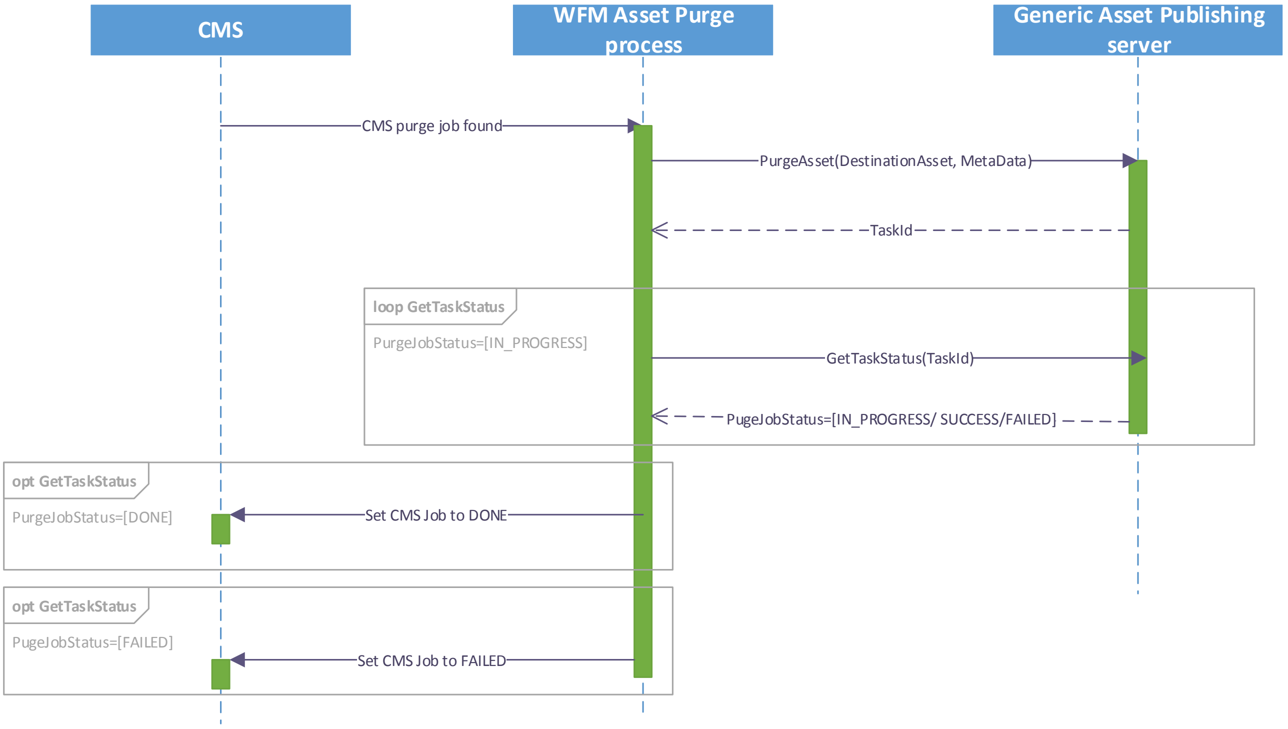


Figure 3: Basic call sequence for PurgeAsset

Purge asset task state diagram

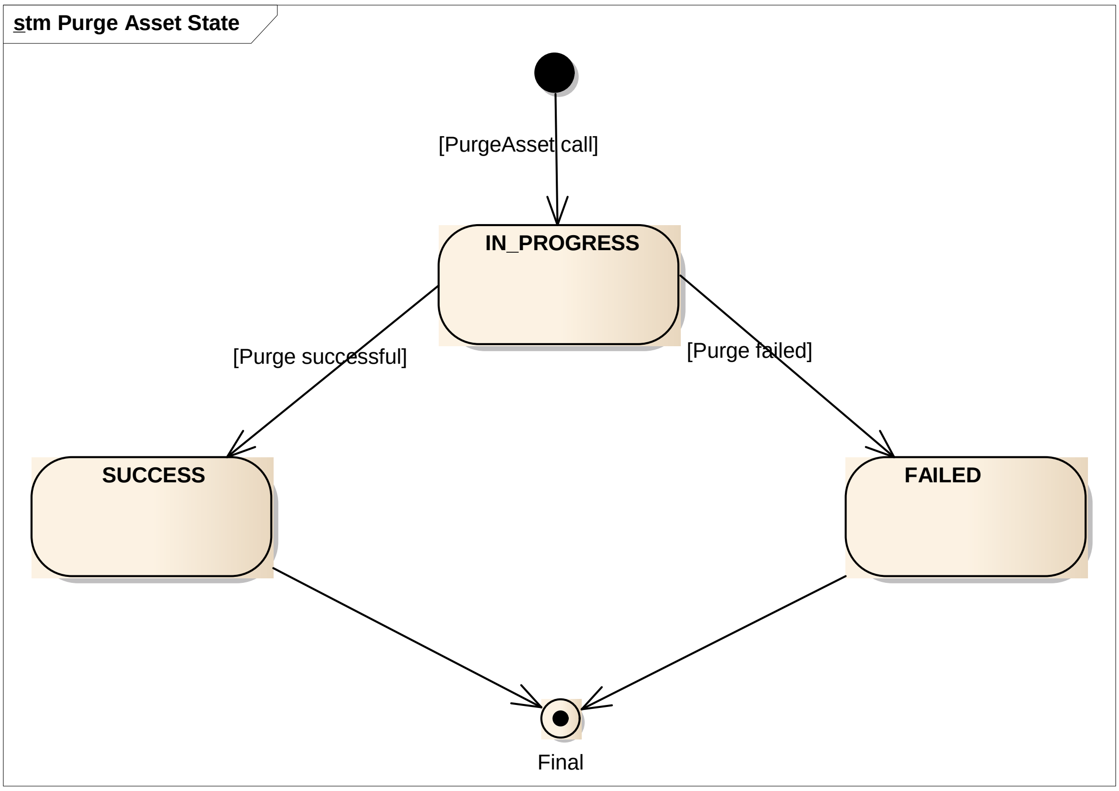


Figure 2: Purge Asset State Diagram

1.3 SOAP mechanism

The WSDL (APUBI.wsdl) is provided with xsd schema in order to validate (and type) the SOAP message.

For every SOAP method, a request, a response and a fault xsd are provided.

Note

APUBI.wsdl and all XSDs are available in AssetPublishing-Interface-v1.0.0.zip file and this file is available in share point.

1.3.1 Fault management

The fault xsd only allow validating the “detail” part of the fault message:
See example below in bold:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <env:Body>
      <env:Fault>
         <faultcode>env:Server</faultcode>
         <faultstring>Task Id does not exist </faultstring>
         <detail>
            <ns2:ExceptionWS xmlns:ns2="wfm:APUBI/v1/schemas">
               <message>Task Id does not exist</message>
            </ns2:ExceptionWS>
         </detail>
      </env:Fault>
   </env:Body>
</env:Envelope>
```

Find below its definition:

The “Fault” sections of this document list error messages that must at least be managed.

It is possible for the publisher to specify additional error messages with the only constraint that they must be human readable.

Note

No specific behavior is planned to be done on error return on CMS side so no error code is managed.

2. Interface specification

2.1 LoadAsset

2.1.1 Description

The goal of this method of interface is to publish an asset in the publishing system.

Nagra CMS constructs a request message containing "SourceAssetURL", "MetaData" and also "DestinationAsset". Response message consists of unique "TaskId" of the operation. Nagra CMS constructs a default URL for the "DestinationAsset" and sends as part of request message.

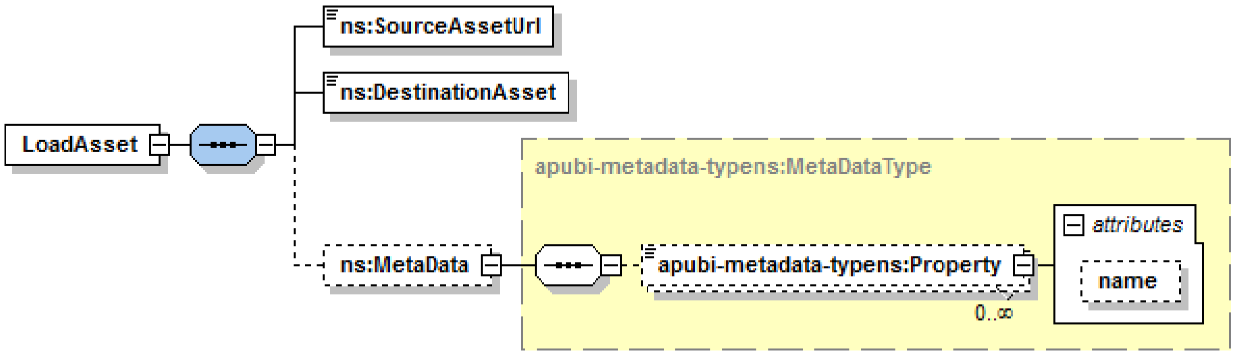
This value can be overridden by "DestinationAsset" value of GetTaskStatus response if the status is 'SUCCESS' and

- If third party publishes at a different location than the URL provided as part of request

- In cases where third party CDNs doesn't disclose the published location of the asset then a unique identifier of the asset should be sent. Nagra CMS uses the same unique identifier while issuing any subsequent purge request.

2.1.2 LoadAsset Request

2.1.2.1 LoadAsset Xsd

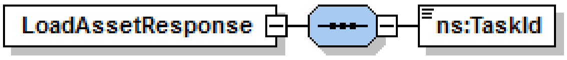


2.1.2.2 Parameters description

Name	Type	Description
SourceAssetUrl	xs:anyURI	URL of the source asset to publish into destination CDN. The asset may be a single file (eg: *.ts) or a directory. If the asset is single file then the SourceAssetURL contains complete path of the file e.g., ftp://user:password@CIS:21/repository/movie.ts If the asset is a directory then the SourceAssetURL contains the path as follows: ftp://user:password@CIS:21/repository/
DestinationAsset	xs:string	Expected URL of where the asset should be published. However it is not mandatory for 3rd party to publish the asset on this location as long as an identifier for the asset or the location of the asset, if different, is sent back in the GetTaskStatus's Response when the status is 'SUCCESS'. The asset may be a single file (eg: *.ts) or a directory. If the asset is single file then the DestinationAsset contains complete path of the file e.g., ftp://user:password@CDN:21/data/cdn/movie_01072015100045.ts If the asset is a directory then the SourceAssetURL contains the path of the as follows: ftp://user:password@CDN:21/data/cdn/
MetaData	xs:MetaDataType	Refer section 2.4 for MetaData details

2.1.3 LoadAsset Response

2.1.3.1 LoadAssetResponse Xsd



2.1.3.2 Parameters description

Name	Type	Description
TaskId	xs:string	ID of the publishing job as created by the publishing system. The format of the "TaskId" parameter is dependent on the publishing system. This ID shall be used by Nagra CMS to query the status of the publishing Job.

2.1.4 Fault

Third party publishing server may throw any error which occurred during the execution of publish job as fault message. Below are a few examples for failure cases:

- Source asset does not exist
- Source asset URL not accessible
- Any internal error which occurred during publish job execution

See 1.3.1 for a fault response sample.

2.1.5 Samples

2.1.5.1 Use case

2.1.5.1.1 Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:LoadAsset>
            <ns:SourceAssetUrl>ftp://user:password@CIS:21/repository/movie.ts</ns:SourceAssetUrl>
            <ns:DestinationAsset>ftp://user:password@CDN:21/data/cdn/movie_01072015100045
            </ns:DestinationAsset >
            <ns:MetaData>
                <ns:Property name=”AssetId”>LYS00001234</ns:Property >
            </ns:MetaData>
        </ns: LoadAsset >
    </soapenv:Body>
</soapenv:Envelope>
```

2.1.5.1.2 Response

Success scenario:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:LoadAssetResponse>
            <ns:TaskId>110E8400-E29B-11D4-A716-446655440000</ns:TaskId>
        </ns: LoadAssetResponse >
    </soapenv:Body>
</soapenv:Envelope>
```

Failure scenario:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>env:Server</faultcode>
            <faultstring>Source asset does not exist </faultstring>
            <detail>
                <ns2:ExceptionWS xmlns:ns2="wfm:APUBI/v1/schemas">
                    <message> Source asset does not exist </message>
                </ns2:ExceptionWS>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>
```

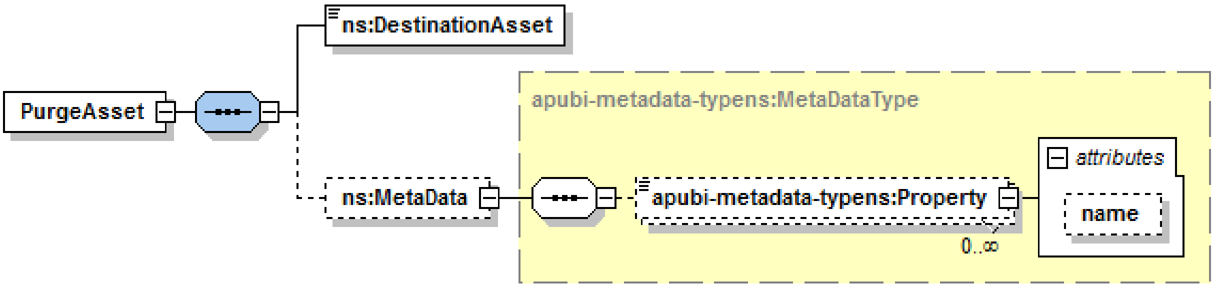
2.2 PurgeAsset

2.2.1 Description

The goal of this interface is to purge an asset on the destination CDN. Whenever a CMS purge job found, the Workflow Manager constructs a request message containing "DestinationAsset" and "MetaData". Response message consists of unique "TaskId" of the operation.

2.2.2 PurgeAsset Request

2.2.2.1 PurgeAsset Xsd


ns:DestinationAsset
PurgeAsset
apubi-metadata-typens:MetaDataType
attributes
ns:MetaData
apubi-metadata-typens:Property
name
0..∞

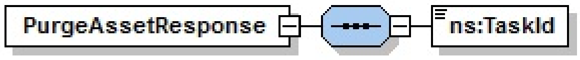
2.2.2.2 Parameters description

Name	Type	Description
DestinationAsset	xs:string	URI/Id of the destination repository from where the asset will be purged. The asset may be a single file (eg: *.ts) or a directory. If the asset is single file then the DestinationAsset contains complete path of the file e.g., ftp://user:password@CDN:21/data/cdn/movie_01072015102030 If the asset is a directory then the DestinationAsset contains the path of the as follows: ftp://user:password@CDN:21/data/cdn/ Or if the DestinationAsset is an identifier which points the asset on the CDN Eg., movie_01072015102030

Name	Type	Description
MetaData	xs:MetaDataType	Refer section 2.4 for MetaData details.

2.2.3 PurgeAsset Response

2.2.3.1 PurgeAssetResponse Xsd



2.2.3.2 Parameters description

Name	Type	Description
TaskId	xs:string	ID of the publishing job as created by the publishing system. The format of the "TaskId" parameter is dependent on the publishing system.

2.2.4 Fault

Third party publishing server may throw any error which occurred during the execution of purge job as fault message. Below are a few examples for failure cases:

- Destination asset URI not accessible
- Not able to delete Destination asset
- Any internal error which occurred during purge job execution

See 1.3.1 for a fault response sample.

2.2.5 Samples

2.2.5.1 Use case

2.2.5.1.1 Request

For a single file:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:PurgeAsset>
            <ns:DestinationAsset>ftp:// user:password@CDN:21/movie_01072015102030.ts
            </ns:DestinationAsset>
            <ns:MetaData>
                <ns:Property name="AssetId" >LYS00001234</ns:Property >
            </ns:MetaData>
        </ns: PurgeAsset >
    </soapenv:Body>
</soapenv:Envelope>
```

For a directory:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:PurgeAsset>
            <ns:DestinationAsset>ftp://user:password@CDN:21/data/cdn/
            </ns:DestinationAsset>
            <ns:MetaData>
                <ns:Property name="AssetId" >LYS00001234</ns:Property >
            </ns:MetaData>
        </ns: PurgeAsset >
    </soapenv:Body>
</soapenv:Envelope>
```

2.2.5.1.2 Response

Success scenario:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:PurgeAssetResponse>
            <ns:TaskId>110F8400-E29B-11D4-A716-446655440</ns:TaskId>
        </ns:PurgeAssetResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Failure scenario:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>env:Server</faultcode>
            <faultstring>Destination asset does not exist </faultstring>
            <detail>
                <ns2:ExceptionWS xmlns:ns2="wfm:APUBI/v1/schemas">
                    <message> Destination asset does not exist </message>
                </ns2:ExceptionWS>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>
```

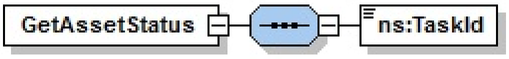
2.3 GetTaskStatus

2.3.1 Description

The goal of this interface is to retrieve the publish/purge task status.

2.3.2 GetTaskStatus Request

2.3.2.1 GetTaskStatus Xsd

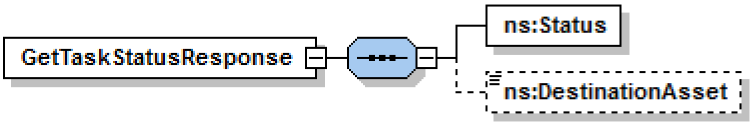


2.3.2.2 Parameters description

Name	Type	Description
TaskId	xs:string	ID of the publishing job returned by "LoadAsset" method Or ID of the purge job returned by "PurgeAsset" method.

2.3.3 GetTaskStatus Response

2.3.3.1 GetTaskStatusResponse Xsd



2.3.3.2 Parameters description

Name	Type	Description
Status	xs:StatusType	SUCCESS: The specified publishing job is successfully completed FAILED: The specified publishing job has failed IN_PROGRESS The specified publishing job is in progress
DestinationAsset	xs:string	URI/Id of the destination repository from where the asset has been published. If the task status is 'SUCCESS' for LoadAsset, then the 'DestinationAsset' must have URI/Id for the destination file or directory as part of GetTaskStatus response. In case of PurgeAsset, GetTaskStatus contains any value for 'DestinationAsset' as part of response then the value will be ignored. The asset may be a single file (eg: *.ts) or a directory. If the asset is single file then the 3rd party may return DestinationAsset as part of response which contains complete path of the file e.g., ftp://user:password@CDN:21/data/cdn/movie_01072015100045.ts If the asset is a directory then the 3rd party may return DestinationAsset as part of response which contains the path of the as follows: ftp://user:password@CDN:21/data/cdn/ Or the 3rd party may return an identifier as DestinationAsset as part of response which points the asset on the CDN. Eg., movie_01072015100045

2.3.4 Fault

Third party publishing server may throw any error which occurred while getting the status of publish/purge as fault message. Below are a few examples for failure cases:

- TaskId does not exist
- Any internal error which occurred while getting the publish/purge job status

See 1.3.1 for a fault response sample.

2.3.5 Sample

2.3.5.1 Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <ns:GetTaskStatus>
         <ns:TaskId>110E8400-E29B-11D4-A716-446655440000</ns: TaskId>
      </ns:GetTaskStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

2.3.5.2 Response

Success scenario:

Sample1:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <ns: GetTaskStatusResponse>
          <ns:Status>IN_PROGRESS</ns: Status >
      </ns: GetTaskStatusResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Sample2:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <ns: GetTaskStatusResponse>
          <ns:Status>SUCCESS</ns: Status >
          <ns:DestinationAsset>http://user:password@CDN:21/ movie_01072015100045.ts
          </ns:DestinationAsset >
      </ns: GetTaskStatusResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Failure scenario:

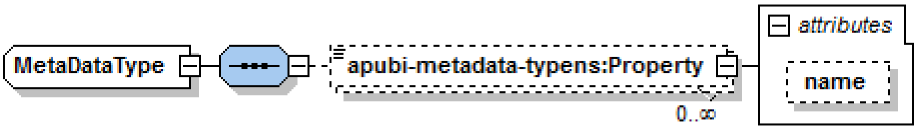
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="wfm:APUBI/v1/schemas">
    <soapenv:Header/>
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>env:Server</faultcode>
            <faultstring>Task Id does not exist </faultstring>
            <detail>
                <ns2:ExceptionWS xmlns:ns2="wfm:APUBI/v1/schemas">
                    <message>Task Id does not exist</message>
                 </ns2:ExceptionWS>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>
```

2.4 MetaData details

MetaData contains the properties which contains the information specific to asset (e.g., AssetId, AssetName etc.) or any other info. The value of the property corresponds to the element body.

2.4.1 Description

2.4.1.1 MetadataType Xsd



2.4.1.2 Parameters description

<Property> attributes details		
Name	**Type**	**Description and typical value**
name	xs:string	Name specifies the data field whose value is carried in the payload of this element.

Sample:

```
<ns:MetaData>
        <ns:Property name=”AssetId” >LYS00001234</ns:Property >
         <ns:Property name=”AssetName”> CMS4X001_SD.ts</ns:Property >
</ns:MetaData>
```

Acronyms

Acronym	Stands for	Definition
CDN	Content Delivery Network	Storage Server dedicated to deliver content to end user devices
CMS	Content Management System	Backend system managing content, traffic and scheduling data
ID	Identifier	Unique value associated to an element to identify it.
URL	Uniform Resource Locator	Address of a file (resource) accessible on the Internet.
URI	Uniform Resource Identifier	ID or Address of a file (resource).
XML	eXtensible Markup Language	XML is standard language used to create extensible data structures (XML schemas) and documents compatible with these structures (XML documents). Allows designers to create their own customized tags, enabling the definition, transmission, validation, and i
WFM	Workflow Manager	Part of the CMS in charge of all external system interactions.

-End of document-