



Content Management Interface

Approved Version 1.0 – 05 July 2011

Open Mobile Alliance
OMA-TS-CMI-V1_0-20110705-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE.....	5
2. REFERENCES	6
2.1 NORMATIVE REFERENCES	6
2.2 INFORMATIVE REFERENCES	6
3. TERMINOLOGY AND CONVENTIONS.....	7
3.1 CONVENTIONS	7
3.2 DEFINITIONS.....	7
3.3 ABBREVIATIONS	7
4. INTRODUCTION	8
4.1 VERSION 1.0	8
5. CMI PROTOCOL	9
5.1 INTERACTION MODEL.....	9
5.1.1 General Response Message Structures.....	10
5.2 PROTOCOL CHOICE	11
5.2.1 Protocol Binding	11
5.2.2 HTTP Binding.....	11
5.3 DATA TYPES	12
5.4 CASE SENSITIVITY	12
5.5 CONTENT MANAGEMENT FUNCTIONALITY	12
5.5.1 Content Upload.....	12
5.6 PURCHASE CONTROL FUNCTIONALITY.....	16
5.6.1 Purchase Operation	16
5.6.2 MakeUnavailable / MakeAvailable Operations	17
5.7 SERVICE METRICS FUNCTIONALITY	18
5.7.1 Report Query Operations	18
5.8 STATUS CODES	19
APPENDIX A. CHANGE HISTORY (INFORMATIVE).....	20
A.1 APPROVED VERSION HISTORY	20
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....	21
B.1 SCR FOR CMI REQUESTS.....	21
B.2 SCR FOR CMC.....	21
B.3 SCR FOR BPC	21
APPENDIX C. REQUEST EXAMPLES (INFORMATIVE).....	22
C.1 SKELETON OF CMI REQUEST USING WEB SERVICE BINDING	22
C.2 SKELETON OF CMI REQUEST USING HTTP BINDING	22
C.3 BODY OF CMI UPLOAD REQUEST – DECOUPLED UPLOAD	22
C.4 BODY OF CMI UPLOAD REQUEST – SELF-CONTAINED UPLOAD.....	23
C.5 BODY OF CMI REPORT QUERY REQUEST	24
C.6 BODY OF CMI PURCHASE REQUEST	24
C.7 BODY OF CMI MAKEUNAVAILABLE REQUEST	24
C.8 BODY OF CMI MAKEAVAILABLE REQUEST	24
APPENDIX D. RESPONSE EXAMPLES (INFORMATIVE).....	25
D.1 SKELETON OF SUCCESS RESPONSE REUSING HTTP LAYER (WEB SERVICES OR HTTP BINDING)	25
D.2 SKELETON OF CMI RESPONSE USING WEB SERVICE BINDING.....	25
D.3 SKELETON OF CMI RESPONSE USING HTTP BINDING	25
D.4 BODY OF CMI REPORT QUERY RESPONSE.....	25

Figures

Figure 1 CMI Request Response Model.....	9
Figure 2 Self-contained Upload	13
Figure 3 - Decoupled Upload	13

Tables

Table 1 Success Response Message	10
Table 2 - Error Response Message.....	11
Table 3 - Content-item-status structure.....	11
Table 4 - CMI Mapping to DCD.....	12
Table 5 - Content-Package structure	14
Table 5 - Content-Item structure.....	14
Table 6 - Content Metadata structure	15
Table 7 Content Upload Request message.....	16
Table 8 - Purchase Request.....	17
Table 9 Purchase Response - extended parameters added to General Response structures.....	17
Table 10 MakeUnavailable / MakeAvailable Request.....	17
Table 11- ReportQuery Request.....	18
Table 12 - ReportQuery Response.....	18
Table 13 - Report Descriptor Structure.....	18
Table 14 - Status Codes	19

1. Scope

The CMI enabler provides mechanisms for the management of content by Content Providers and Service Providers. This specification defines the API between Content Provider and Service Provider (e.g. Operator) for managing content. It covers the following areas:

- Purchase Content Activities
- Download/Upload Content
- Statistical Content Information

The implementation of the API by the Service Providers and the Content Provider functionalities are out of the scope of this specification.

2. References

2.1 Normative References

[CMI-AD]	“Content Management Interface Architecture”, Open Mobile Alliance™, OMA-AD-CMI-V1_0, URL: http://www.openmobilealliance.org/
[CMI-RD]	“Content Management Interface Requirements”, Open Mobile Alliance™, OMA-RD-CMI-V1_0, URL: http://www.openmobilealliance.org/
[CMI-XSD]	“CMI 1.0 XML Schema”, Open Mobile Alliance™, OMA-SUP-XSD_cmi_package-V1_0, URL: http://www.openmobilealliance.org
[DCD-TS]	“Dynamic Content Delivery”, Open Mobile Alliance™, OMA-TS-DCD_Semantics-V1_0 URL: http://www.openmobilealliance.org/
[HTTP]	HTTP v1.1 – RFC 2616, URL: http://www.w3.org/Protocols/rfc2616/rfc2616.html
[FTP]	File Transfer Protocol – RFC 959, URL: http://www.w3.org/Protocols/rfc959/
[ISO8601]	“Data elements and interchange formats -- Information interchange -- Representation of dates and times”, International Organization for Standardization (ISO), URL: http://www.iso.org/
[RFC2046]	MIME Part 2, Media Types – RFC 2046, URL: http://www.ietf.org/rfc/rfc2046.txt
[RFC2119]	“Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL: http://www.ietf.org/rfc/rfc2119.txt
[RFC4281]	The Codecs Parameter for "Bucket" Media Types – RFC 4281 URL : http://www.ietf.org/rfc/rfc4281.txt
[SOAP1.1]	“Simple Object Access Protocol (SOAP) 1.1”, May 8, 2000, URL: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
[SOAP1.2]	“Simple Object Access Protocol (SOAP) 1.2”, April 27, 2007, URL: http://www.w3.org/TR/soap12-part1/
[SCRRULES]	“SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL: http://www.openmobilealliance.org/
[XML Schema Part 2: Datatypes]	“XML Schema Part 2: Datatypes”, W3C Recommendation 02 May 2001, URL: http://www.w3.org/

2.2 Informative References

[CMR]	“Customized Multimedia Enabler Release Definition”, Version 1.0, Open Mobile Alliance™, OMA-ERELED-CMR-V1_0, URL: http://www.openmobilealliance.org/
[FCC TV Parental Guidelines]	The TV Parental Guidelines, URL: http://www.fcc.gov
[OMADICT]	“Dictionary for OMA Specifications”, Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-Vx_y, URL: http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Business Process Component	The entity that exposes CMI-2, CMI-3 and CMI-4 interfaces defined in [CMI-AD].
Content Item	A single unit of content delivered from Content Provider to Service Provider. Content Item consists of Content Metadata and content payload, the latter is opaque for the CMI enabler. The schema for Content Item is defined in [CMI-XSD].
Content Management Component	The entity that exposes CMI-1 interface defined in [CMI-AD].
Content Metadata	Attributes and child elements of a Content Item. The schema for Content Metadata is defined in [CMI-XSD].
Content Package	A bundle of one or more Content Items. The schema for Content Package is defined in [CMI-XSD].
Content Provider	See [CMI-RD]
Decoupled Upload	An upload operation in which sending upload request and uploading Content Package are decoupled and performed separately (e.g. content upload over FTP protocol and upload request over HTTP).
Network Operator	See [OMADICT]
Self-contained Upload	An upload operation in which Content Package is included in the upload request (e.g. using HTTP PUT)
Service Provider	See [OMADICT]

3.3 Abbreviations

API	Application Programming Interface
BPC	Business Process Component
CMI	Content Management Interface
CMC	Content Management Component
CMR	Customized Multimedia Ringing
CP	Content Provider
CPDE	Content Provider DCD Enabled
DCD	Dynamic Content Delivery
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol
SP	Service Provider

4. Introduction

The CMI Enabler supports the content management needs of Service Providers (e.g. Network Operators, Web Portal Service Providers, or other Value-Added Service Providers) and Content Providers, as they deliver content-related services, e.g. Operator-provided services such as Customized Multimedia Ringing (CMR) [CMR], Mobile Advertisement, etc. Use of the term “services” in this document refers to such services offered by Service Providers and Content Providers.

These services require multiple interactions between the systems operated by the Service Provider and the Content Provider, for the overall purpose referred to in CMI as *content management*. While “content management” in specific contexts can refer to a broad set of functionality, e.g. management of work flows related to creation, storage, and publication of digital media and electronic text, content management in the context of OMA CMI means specifically:

- Coordinated actions across an interface between two entities, described as the Content Provider (source of the content) and Service Provider (provider of services within which the content is made available)
- Across that interface, actions related to specific content items and their use in services such as:
 - Upload content, primarily from the Content Provider to the Service Provider. This is also known as content ingestion.
 - Manage the uploaded content, e.g. remove or change its attributes, etc.
 - Purchasing actions: Purchase content, including selection of purchase terms, e.g. content price, etc.
 - Statistics and report information: Content usage information, etc.

The CMI Enabler supports the deployment of services that depend upon such interactions via a standardized set of functions and interfaces, improving service deployment flexibility, interoperability, cost, and time-to-market. This will further enable consistent deployment of services using a variety of models for supplying content, including “on-deck” and “off-deck” as described in [CMI-RD].

4.1 Version 1.0

Version 1.0 of the CMI Enabler addresses all of the functional requirements included in [CMI-RD] for CMI 1.0.

5. CMI Protocol

5.1 Interaction Model

Figure 1 provides a high level overview of the interaction between a Content Provider and a Service Provider.

In the following diagram, solid lines represent mandatory messages, while dashed line represent message that are either optional or conditional.

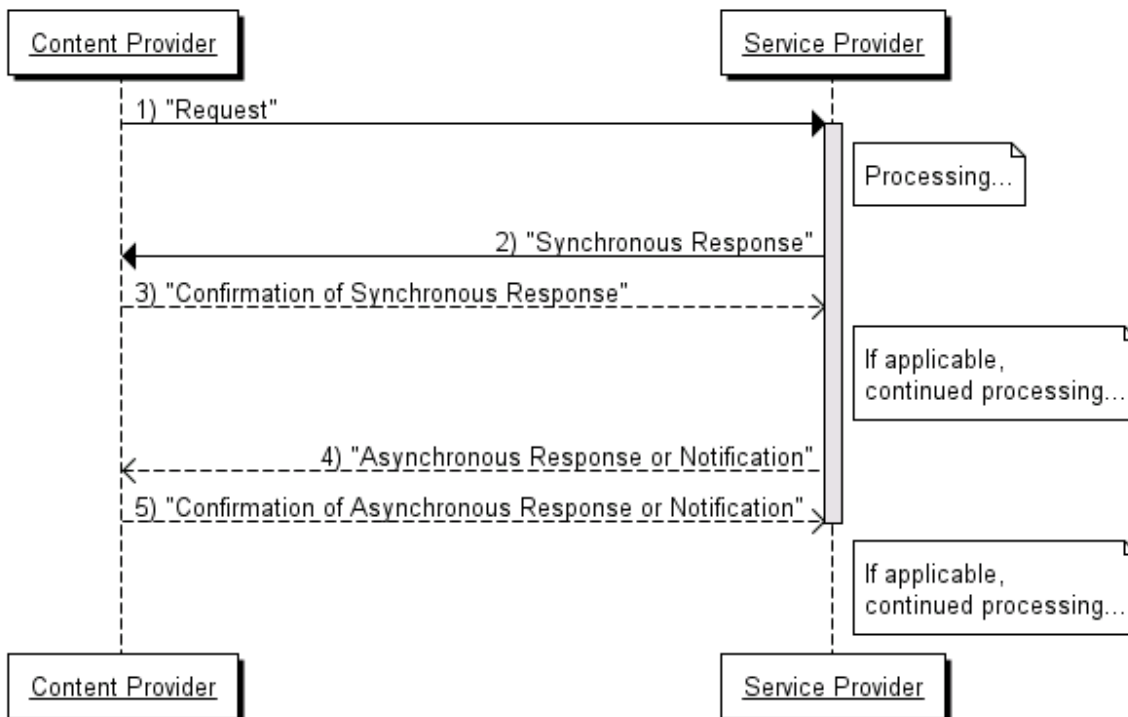


Figure 1 CMI Request Response Model

The communication between the Content Provider and the Service Provider SHALL be based on a request-response operation model, that is, a CMI operation consists of a request (1) followed by one or more responses (2) and (4) with optional confirmation of the response (3) and (5). A “response” in this case refers to actions taken by the Service Provider as part of a dialog initiated by the request, which may include:

- one or more synchronous responses messages (2), i.e. occurring within the transport protocol connection of the original request (1)
- one or more asynchronous responses (4) within the context of the CMI operation, i.e. occurring within a new transport protocol connection, but related to the original request (1) by a CMI protocol operation identifier
- one or more asynchronous notifications (4), i.e. while initiated by the original request (1), not specifically related to it by CMI protocol operation identifier

A response may be optionally confirmed ((3) or (5)) by the Content Provider, if applicable to the operation.

An operation is closed only when all responses have been received and confirmed, if required. For example the Content Provider submits a request to the Service Provider. The Service Provider receives the request, parses it, handles it and generates a closing response to the originating Content Provider.

A request may require additional processing by the Service Provider, e.g. service metrics collection, content validation, content status checking.

Of these interaction types described above, the synchronous request/response SHALL be supported.

Note:

Figure 1 shows two separate logical entities – The Content Provider and the Service Provider. These logical entities can be deployed as single physical entity.

While many response messages have common structure, request messages differ significantly. The structures for request messages will be defined in the following sections.

5.1.1 General Response Message Structures

This section specifies reusable message structures for success and for error responses.

Unless specified otherwise, response messages (both synchronous and asynchronous) defined in this specification reuse these general structures and MAY include possible extension parameters. These extension parameters are added at the end of the general structures.

5.1.1.1 Success Response

This message indicates that the request has been successfully processed or still pending validation.

In response to the request, the CMC SHALL respond with a protocol layer response, e.g. HTTP 200 OK.

When it is required to notify the CP about pending validation of the request by the CMC, the response sent by the CMC SHOULD include a CMI success response message, as described in Table 1. This behavior depends upon the nature of the service (e.g. CMR), type of the operation (e.g. Purchase), or Service Provider's Policy

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Conditional: Mandatory if the request had a Operation-id, in which case the value is identical to the Operation-id sent in the request.
Validation-pending	A	1	Boolean	Indicates whether CMC completed processing of the request payload or additional validation is pending. Default value: False (processing completed)
Additional-info	E	0..1	String	Optional message to CP (e.g. description of validation step).

Table 1 Success Response Message

5.1.1.2 Error Response

Error response indicates that at least one error encountered in processing the corresponding request, and provides details of the error(s).

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Conditional: Mandatory if the request had a Operation-id, in which case the value is identical to the Operation-id sent in the request.
Content-item-status	E	1..n	List of Structures	Description of the processing result of each content item. The CMC MUST include in this list a single structure for each Content-Item which failed processing.

				The CMC MAY include in this list a single structure for each Content-Items which is capable for successful processing, in order to provide detailed status. Content-Items that were included in the request and are not reported in this list are considered capable for successful processing.
Success-items-accepted	E	0..1	Boolean	TRUE value indicates that only the Content-Items whose reported status-code in the Content-item-status list indicated error were rejected and others were successfully processed by the CMC. FALSE value indicates that all the Content-Items in the request were rejected due to the overall errors found within the request. Default value: FALSE

Table 2 - Error Response Message

Name	Attribute/Element	Cardinality	Data Type	Description
Content-id	E	1	String	Identifier set by the Content Provider, e.g. UUID. Matches the Content-id which was submitted in the upload request.
Status-code	E	1	Non Negative Integer	Status code as defined in section 5.8.
Status-text	E	0..1	String	Status description

Table 3 - Content-item-status structure

5.2 Protocol Choice

5.2.1 Protocol Binding

The CMC and BPC SHALL support at least one of the following bindings.

5.2.1.1 Web Services Binding

The web-service binding between the Content Provider and the Service Provider SHALL rely on [SOAP1.1] or [SOAP1.2] over HTTP(S) [HTTP].

The SOAP request SHALL be sent as the body of a HTTP(S) POST request, as defined in [SOAP1.1] or [SOAP1.2].

The SOAP response SHALL be sent as a body of the HTTP(S) response.

Both the request and the response SHALL contain a single SOAP envelope, conformant with [SOAP1.1] or [SOAP1.2].

5.2.2 HTTP Binding

The HTTP binding between the Content Provider and the Service Provider SHALL rely on HTTP(S) [HTTP].

5.2.2.1 CMI Mapping to DCD

CMI Enabler could be used by the Service Provider to receive content that will be delivered to mobile devices using DCD Enabler. In this case CMI operations can replace some of operations over DCD-CPDE interface. When using CMI to upload DCD content items, the following requirements apply:

- The Content Upload Request message defined in CMI (see Section 5.5.1) corresponds to DCD-CPDE ContentUpdate message [DCD TS] and SHALL be sent as an XML document in the body of a HTTP(S) POST request [HTTP]. This XML document SHALL contain DCD namespace reference and location of the DCD XML schema [DCD XSD].

- Content element within Content Package SHALL be formatted as Server-Package element defined in the [DCD XSD].
- Message parameters and content metadata elements (section **Error! Reference source not found.**) SHALL be mapped as follows:

CMI metadata parameter	Corresponding parameter in [DCD TS]
Package-id	Content-block-id
Service-id	Channel identifier
Content-reference	Content-address
Tags	Content-types

Table 4 - CMI Mapping to DCD

5.3 Data Types

The datatypes of the CMI parameters used in this specification are defined in the section “Built-in datatypes” of [XML Schema Part 2: Datatypes] specification by the W3C, as adapted by SOAP 1.1. The datatypes discussed there are computer representations of well known abstract concepts such as integer and date.

Following are informative examples of built-in datatypes. For their full definition refer to [XML Schema Part 2: Datatypes]:

- string represents character strings in XML, i.e. the set of finite-length sequences of characters.
- token represents tokenized strings, i.e. strings that do not contain the line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20) and that have no internal sequences of two or more spaces.

Wherever applicable, CMI implementations SHALL use the canonical lexical representation of these datatypes, as defined in [XML Schema Part 2: Datatypes].

Following are informative examples of the canonical lexical representation of built-in datatypes. For their full definition refer to [XML Schema Part 2: Datatypes]:

- The canonical representation for integer is defined by prohibiting certain options from the Lexical representation. Specifically, the preceding optional “+” sign is prohibited and leading zeroes are prohibited: so ‘+00012’ becomes ‘12’
- The canonical representation for unsignedLong is defined by prohibiting certain options from the Lexical representation. Specifically, leading zeroes are prohibited: so ‘0000321323’ becomes ‘321323’

5.4 Case Sensitivity

XML element names (XML tags) as well as values of XML enumeration types SHALL be case sensitive (this is enforceable by XML schema validation).

5.5 Content Management Functionality

The Content management functionality described in this section corresponds to CMI-1 interface in the [CMI-AD].

5.5.1 Content Upload

The Content Upload operation allows a Content Provider to submit new content to Service Provider. It is also used to replace existing content.

Uploading a single Content Item or a collection of Content Items is performed in the same manner, hence reusing the same operation syntax and semantics.

The Content Items (including associated Content Metadata) can be combined in a Content Package as defined in section 5.5.1.1. Content-Items within the Content Package consist of Content Metadata and content payload, the latter is opaque to the CMI enabler. Alternatively, content payload MAY be referenced by Content-reference pointing to the content location (instead of using the Content element).

The Content Metadata is formatted as defined in section 5.5.1.2.

Content Upload could be performed using any combination of HTTP and FTP transport protocols. Depending on the protocols used, the Content Upload operation could be performed either as a Self-contained Upload or as a Decoupled Upload.

In Self-contained Upload the Content Package is embedded within the Content Upload Request message.

Self-contained Upload operation is shown in Figure 2.

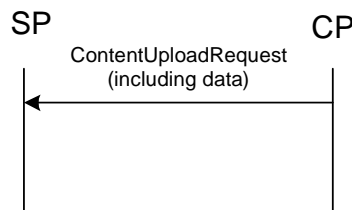


Figure 2 Self-contained Upload

In Decoupled Upload the Content Package is uploaded separately and decoupled from the ContentUpload Request message, and the Package-id is identified in the Content Upload Request.

Decoupled Upload operation is shown in Figure 3.

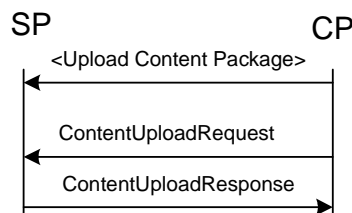


Figure 3 - Decoupled Upload

In case errors are encountered when processing the request, the CMC SHALL respond to the request with error information.

Upon receiving error response, the CP can correct and resubmit failed content items to the SP. Upon receiving the confirmation of delivery (if applicable) for error response, the CMC should discard failed content items

5.5.1.1 Content Package

A Content Package element SHALL contain one or more Content Items.

Each Content-Item SHALL contain Content Metadata and MAY contain content payload, parameters specific for the Content Package and collection of Content Metadata for the Content Items in the package.

The Content Package, Content Item, and Content Metadata SHALL comply with the format defined by [CMI-XSD].

When using Self-contained Upload, the Content-Package element SHALL be included in the Content Upload Request message.

When using Decoupled Upload, the Content Package SHALL be uniquely identified by a Package-id included in the Upload Request message which is used for the correlation of the request with the Content-Package. In this case, the ContentPackage SHALL be the root element in an XML file whose filename SHALL be identical to the Package-id with '.xml' as extension.

Name	Attribute/Element	Cardinality	Data Type	Description
Content-item	E	1..n	List of Structures	Root element for the bundle of Content-Items

Table 5 - Content-Package structure

Name	Attribute/Element	Cardinality	Data Type	Description
Content-metadata	E	1	Structure	Content Metadata for the Content Item.
Content	E	0..1	Opaque Data	Content payload. Opaque for the CMI enabler. If it is not present in the Content Item it SHALL be referred using Content-reference parameter in Content Metadata.
Entry-index	A	0..1	Positive Integer	A sequential number used to enumerate content items in the package for human-readability, allowing easily reference a specific record (e.g. in error response to Content Provider).

Table 6 - Content-Item structure

Name	Attribute/Element	Cardinality	Data Type	Description
Content-id	A	1	String	Content item identifier specified by the Content Provider (e.g. UUID). This parameter SHOULD be unique within the Service Provider domain and SHALL be unique within the Content Package.
Replaces-content-id	A	0..1	anyURI	Content identifier of previously uploaded outdated content item that needs to be replaced with the current content item. Subject to Service Provider policy, some services may not require to support this attribute.
Content-block-id	A	0..1	String	Identifies which multiple Content Items can be associated as a block (i.e. a group of related Content Items). Can be used by the Service Provider for content aggregation / bundling. This attribute MAY be identical to Content Package identifier used in ContentUploadRequest message.
Content-name	E	0..n	String	Name(s) of content item in a human readable format, possibly in multiple languages. The language of this element can be expressed using built-in XML attribute xml:lang with this element.
Author	E	0..n	List of Strings	Author(s) for the content item, if appropriate (e.g. "Joe Doe", "Disney", etc.)
Content-reference	E	0..1	anyURI	Location of the content payload, if appropriate. SHALL be provided if associated content payload wasn't provided with the metadata.
MIME-type	E	1	String	Mime (media) type of the content item (see [RFC2046]). Could include media codec parameters where applicable, as per [RFC4281].
Content-length	A	0..1	Integer	The size in bytes of the content item. Subject to Service Provider policy, some services may not require to support this attribute.
Encoding	A	0..1	String	Encoding that has been applied to the content item, e.g. "gzip", "deflate", "base64".

Updated	A	0..1	dateTime	Time when the content item was last updated. SHALL conform to the "date-time" definition in [ISO8601]. In addition, an uppercase "T" character SHALL be used to separate date and time, and an uppercase "Z" character SHALL be present.
Self-expiration	E	0..1	Positive Integer	Default number of days that the content is to be available for use by the purchaser, from the date of purchase, subject to the Content-expiration parameter. Subject to Service Provider policy, some services may not require to support this attribute.
Content-expiration	A	0..1	dateTime	Time when the content item should expire from Service Provider storage, or no longer be made available for related services. SHALL conform to the "date-time" definition in [ISO8601]. In addition, an uppercase "T" character SHALL be used to separate date and time, and an uppercase "Z" character SHALL be present.
Tags	E	0..n	List of Strings	A list of strings that describe the content to enable association or filtering.
Content-price	A	0..1	String	Indicates the price (amount and currency) of this content item. Subject to Service Provider policy, some services may not require to support this attribute.
Parental-rating	A	0..1	String	Content rating per [FCC TV Parental Guidelines] or similar local regulatory requirements. Subject to Service Provider policy, some services may not require to support this attribute.

Table 7 - Content Metadata structure

5.5.1.2 Delivery of Content

The CMC SHALL support at least one of Self-contained Upload and Decoupled Upload.

If Decoupled Upload is supported, then the CMC SHALL support [FTP] as a protocol for receiving Content Packages.

If Self-contained Upload is supported, then the CMC SHALL support [HTTP] as a protocol for receiving Content Packages (including content payload or content reference) embedded within the ContentUpload message. If Content Package contains content reference, the content payload has been previously uploaded (using [FTP] or [HTTP]) to the location specified by this reference.

5.5.1.3 Content Upload Request

With Self-contained Upload, the Content Upload Request message is used by the Content Provider to upload Content Package to the Service Provider. With Decoupled Upload, the Content Upload Request message is used by the Content Provider to signal to the Service Provider Component that a Content Package has been submitted and is ready for processing. Content Upload Request message is also used for content update purposes, i.e. to update payload or Content Metadata of existing Content Items that had been previously processed by the CMC.

The Content Upload Request contains the parameters as shown in Table 1.

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Unique identifier for this operation. If used, this identifier is generated by the requestor for tracking and logging purposes.
Content-package	E	0..n	List of Data Structures	One or more content packages. Each content package consists of content items with payload and metadata. Content payload could be empty if content-reference parameter of the content metadata contains the address where the content could be retrieved (e.g. URI). For content previously uploaded as discrete

				content items (i.e. without content metadata) the URI in the Content-Package SHALL refer to a content location. Conditional: Mandatory, if Self-contained Upload. Not present, if Decoupled Upload
Package-id	E	0..n	List of anyURI	Unique identifier(s) (e.g. UUID) identifying the content package(s) associated with the upload request. Used to correlate Content Upload Request with previously uploaded Content Package. Conditional: Mandatory, if Decoupled Upload. Not present, if Self-contained Upload
Subscription-id	E	0..n	List of Strings	List of subscription identifiers. Used to group content subscribers according to subscription preferences (e.g. subscription filters). Present in the message if content upload is targeted to the particular groups of subscribers.
Service-id	A	0..1	String	Identifier of the service to which these Content Packages are to be associated.

Table 8 Content Upload Request message

5.5.1.4 Content Upload Response

The Content Upload Response message (when used) reuses the General Response Message Structures, as defined in section 5.1.1.

5.6 Purchase Control Functionality

This section specifies Content Provider interactions with the Service Provider to fulfill a purchase of content by the end user.

The Purchase control functionality described in this section corresponds to CMI-3 interface in the [CMI-AD].

5.6.1 Purchase Operation

In this operation the Content Provider notifies the Service Provider about a purchase of content by specific subscriber(s), with the option of specifying expiration time (i.e. that the purchase of the content is for limited time).

The metadata of the purchased content can identify the service which is associated with this purchase.

5.6.1.1 Purchase Request

Subject to Service Provider's Policy, the CMC MAY interpret Purchase Request as a trigger for service subscription for the user if the user is not already subscribed to the service.

The Purchase Request contains the parameters as shown in the following table.

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Unique identifier for this operation. If used, this identifier is generated by the requestor for tracking and logging purposes.
Content-provider-id	E	1	String	Unique identifier which identifies the Content Provider to the CMC.
Content-id	E	1..n	String	Identifier set by the Content Provider, e.g. UUID.

				Matches the Content-id which was submitted in the upload request.
Subscriber-id	E	1	String	Opaque string that uniquely identifies the subscriber or group of subscribers who wish to purchase the content. The format of the identifier can depend on the service associated with the content.
Self-expiration	E	0..1	Positive Integer	Number of days that the content is to be available for use by the purchaser, from the date of purchase.

Table 9 - Purchase Request

5.6.1.2 Purchase Response

The Purchase Response message reuses the General Response Message Structures, as defined in section 5.1.1, with additional parameters as shown in the following table.

Name	Attribute/Element	Cardinality	Data Type	Description
First-purchase-flag	E	0..1	Boolean	Indicates if this is the first purchase made by the subscriber. Value of "True" means that the Purchase Request implicitly enrolled this subscriber to the service. The default value is "False".

Table 10 Purchase Response - extended parameters added to General Response structures

5.6.1.3 Extending a Purchase

In order to extend the expiration date of a previously purchased item the Purchase Request message SHALL be used.

5.6.2 MakeUnavailable / MakeAvailable Operations

By default when content items are uploaded, it is expected that the Service Provider will make them available to end users.

The MakeUnavailable operation is used to request that previously uploaded content becomes unavailable to consume, subject to Service Provider's Policy.

The MakeAvailable operation is used to request that previously unavailable content becomes available to consume, subject to Service Provider's Policy.

Both operations use the same parameters as described below.

5.6.2.1 MakeUnavailable / MakeAvailable Request

Both MakeUnavailable Request and MakeAvailable Request use the parameters shown in Table 11:

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Unique identifier for this operation. If used, this identifier is generated by the requestor for tracking and logging purposes.
Content-provider-id	E	1	String	Unique identifier which identifies the Content Provider to the CMC.
Content-id	E	1..n	List of String	Identifier set by the Content Provider, e.g. UUID. Matches the Content-id which was submitted in the upload request.

Table 11 MakeUnavailable / MakeAvailable Request

5.6.2.2 MakeUnavailable / MakeAvailable Response

Both MakeUnavailable Response and MakeAvailable Response reuse the General Response Message Structures, as defined in section 5.1.1.

5.7 Service Metrics Functionality

Service metrics allow a Content Provider to get statistical and/or operational information related to content that it previously uploaded. Such information can include reports about service performance, statistics of content usage by end-users, upcoming content expiration, etc.

The service metrics functionality described in this section corresponds to CMI-4 interface in the [CMI-AD].

5.7.1 Report Query Operations

5.7.1.1 Report Query Request

This request is used in order to query the BPC about available reports. The request uses the parameters shown in Table 12.

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Unique identifier for this operation. If used, this identifier is generated by the requestor for tracking and logging purposes.
Content-provider-id	E	1	String	Unique identifier which identifies the Content Provider to the BPC.

Table 12- ReportQuery Request

5.7.1.2 Report Query Response

This response lists available reports. The response uses the parameters shown in tables 12 and 13 (it does **not** reuse the general response message structure).

Name	Attribute/Element	Cardinality	Data Type	Description
Operation-id	E	0..1	String	Conditional: Mandatory if the request had a Operation-id, in which case the value is identical to the Operation-id sent in the request.
Report-descriptor	E	0..n	Structure	Information about available report(s)

Table 13 - ReportQuery Response

Name	Attribute/Element	Cardinality	Data Type	Description
Name	E	1	String	Name of the report.
URL	E	1	anyURI	Location where the report is available.
Description	E	0..1	String	Description of the report

Table 14 - Report Descriptor Structure

5.8 Status Codes

This section provides the numeric values and text names for status codes used in CMI messages.

Status Code	Status Description
000	success
001	validation pending
002	security error
003	metadata error
004	content not found
005	invalid parameter
006	not allowed
008	invalid service-id
009	content unacceptable
010	temporary failure
011	subscriber-id not found
012	billing error
999	general error
1000-1999	Proprietary status code indicating success
2000-2999	Proprietary status code indicating errors

Table 15 - Status Codes

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-CMI-V1_0-20110705-A	05 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0223-INP_CMI_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for CMI Requests

Item	Function	Reference	Requirement
CMI-REQ-001-M	Content Package	5.5.1.1	

B.2 SCR for CMC

Item	Function	Reference	Requirement
CMI-CMC-001-M	Content Upload	5.5.1	(CMI-CMC-002-O OR CMI-CMC-003-O) AND (CMI-CMC-004-O OR CMI-CMC-005-O)
CMI-CMC-002-O	Web Services Binding	5.2.1	
CMI-CMC-003-O	HTTP Binding	5.2.2	
CMI-CMC-004-O	Self Contained	5.5.1	
CMI-CMC-005-O	Decoupled	5.5.1	

B.3 SCR for BPC

Item	Function	Reference	Requirement
CMI-BPC-001-O	Purchase Control	5.6	
CMI-BPC-002-O	Content Availability	5.6.2	
CMI-BPC-003-O	Service Metrics	5.7	

Appendix C. Request Examples (Informative)

C.1 Skeleton of CMI Request using Web Service Binding

POST /CMIRRequest HTTP/1.1

Host: cmi.example.com:80

Content-Type: text/xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body xmlns:cmiPkg="urn:oma:xml:cmi:package:1.0">
  <!-- CMI Request message element starts here, e.g.PurchaseRequest -->
    ....
  <!-- CMI Request message element closes here, e.g. /PurchaseRequest -->
</soapenv:Body>
</soapenv:Envelope>
```

C.2 Skeleton of CMI Request using HTTP Binding

POST /CMIRRequest HTTP/1.1

Host: cmi.example.com:80

Content-Type: text/xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- CMI Request message element starts here, e.g.PurchaseRequest
  xmlns="http://www.openmobilealliance.org"
  xmlns:cmiPkg="urn:oma:xml:cmi:package:1.0" -->
  ....
<!-- CMI Request message element closes here, e.g.PurchaseRequest >
```

C.3 Body of CMI Upload Request – Decoupled Upload

```
<ContentUploadRequest >
  <Operation-id>123</Operation-id>
  <!-- Content-Package element is not present for Decoupled Upload-->
  <Package-id >file://somelocation/package_1</Package-id>
  <Package-id >file://somelocation/package_2</Package-id>
  <Package-id >file://somelocation/ package_3</Package-id>
  <ext:non-standard-extension> optional extension data</ext:non-standard-extension>
</ContentUploadRequest >
```

C.4 Body of CMI Upload Request – Self-contained Upload

```

<ContentUploadRequest >
  <cmiPkg:Content-package>
    <cmiPkg:Content-item>
      <cmiPkg:Content-metadata Content-id="0x01A0" ...>
        <cmiPkg:Content-reference>http://some.com/location/trailer.mpg</cmiPkg:Content-reference>
        <cmiPkg:Mime-type>video/mpeg</cmiPkg:Mime-type>
      </cmiPkg:Content-metadata>
    </cmiPkg:Content-item>
    <cmiPkg:Content-item>
      <cmiPkg:Content-metadata Content-id="0x01A1" ...>
        <cmiPkg:Content-reference>http://some.com/location/ movie.mpg </cmiPkg:Content-reference>
        <cmiPkg:Mime-type>video/mpeg</cmiPkg:Mime-type>
      </cmiPkg:Content-metadata>
    </cmiPkg:Content-item>
  </cmiPkg:Content-package>
  <cmiPkg:Content-package>
    <cmiPkg:Content-item>
      <cmiPkg:Content-metadata Content-id="0x01B0" .../>
        <cmiPkg:Mime-type> application/atom+xml</cmi:Mime-type>
      </cmiPkg:Content-metadata>
      <cmiPkg:Content>
        <feed xmlns=http://www.w3.org/2005/Atom>
          <title>New Sci-Fi movie released by Hollywood</title>
          <link href="http://www.abc.com/movies/rss/">
          <updated>2009-09-09T11:59:55Z</updated>
          <id>urn:uuid:1724c695-cfb8-4ebb-aaaa-80da344efa6a</id>
          <entry>
            <title>Out of this world</title>
            <summary>Aliens capture Earth, young Chinese girl saves the world... </summary>
            <id>urn:uuid:61a73c80-d399-11d9-b93C-0003939e0af6</id>
            <updated>2009-09-09T11:59:55Z</updated>
          </entry>
          ...
        </feed>
      </cmiPkg:Content>
    </cmiPkg:Content-item >

  </cmiPkg:Content-package>
  <!--Package-id element is not present for Self-contained Upload-->
  <Subscription-id>SubscriptionId_1</Subscription-id>
  <Subscription-id>SubscriptionId_2</Subscription-id>
  <ext:non-standard-extension> optional extension data</ext:non-standard-extension>
</ContentUploadRequest >

```

C.5 Body of CMI Report Query Request

```
<ReportQueryRequest>
  <Operation-id>Operation_231</ Operation-id>
  <Content-provider-id>ContentProvider_1</Content-provider-id>
</ReportQueryRequest>
```

C.6 Body of CMI Purchase Request

```
<PurchaseRequest >
  <Operation-id>124</Operation-id>
  <Content-provider-id >KJG798GJ465GD</Content-provider-id >
  <Content-id >content1</Content-id >
  <Content-id >content2</Content-id >
  <Subscriber-id >user1</Subscriber-id >
</PurchaseRequest >
```

C.7 Body of CMI MakeUnavailable Request

```
<MakeUnavailableRequest>
  <Operation-id>125</Operation-id>
  <Content-provider-id >CP11</Content-provider-id >
  <Content-id >content11</Content-id >
</MakeUnavailableRequest >
```

C.8 Body of CMI MakeAvailable Request

```
<MakeAvailableRequest>
  <Operation-id>126</Operation-id>
  <Content-provider-id >CP11</Content-provider-id >
  <Content-id >content11</Content-id >
</MakeAvailableRequest>
```


Appendix D. Response Examples (Informative)

D.1 Skeleton of Success Response reusing HTTP layer (Web Services or HTTP binding)

HTTP/1.1 200 OK

D.2 Skeleton of CMI Response using Web Service Binding

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2001/12/soap-envelope">
  <soapenv:Body >
    <!-- CMI Response message element starts here, e.g. PurchaseResponse -->
      ....
    <!-- CMI Response message element closes here, e.g./PurchaseResponse-->
  </soapenv:Body>
</soapenv:Envelope>
```

D.3 Skeleton of CMI Response using HTTP Binding

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- CMI Response message element starts here, e.g. PurchaseResponse
-->
....
<!-- CMI Response message element closes here, e.g. /PurchaseResponse -->
```

D.4 Body of CMI Report Query Response

```
<ReportQueryResponse>
  <Operation-id>Operation_231</Operation-id>
  <Report-descriptor>
    <Name>Report-1</Name>
    <URL>http://somelocation/ report_1</URL>
    <Description>this report is about 1 and is useful for 1</Description>
  </Report-descriptor >
  <Report-descriptor>
    <Name>Report-2</Name>
```

```
<URL>http://someslocation/report_2</URL>  
<Description>this report is about 2 and is useful for 2</Description>  
</Report-descriptor >  
</ReportQueryResponse>
```