Dynamic Content Delivery Technical Specification – Semantics and Transactions

Approved Version 1.0 – 05 Jul 2011

Open Mobile Alliance

OMA-TS-DCD_Semantics-V1_0-20110705-A

Contents

Figures

Tables

1. Scope

This document defines the semantics of Dynamic Content Delivery (DCD) Version 1.0, including the session and transaction model, message types and their information elements, status and error codes, and other details.

2. References

2.1 Normative References

[3GPP 23.041 v5.2.0]	"Technical realization of Cell Broadcast Service (CBS)", V5.2.0 (2003-12), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 5) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.041/
[3GPP 23.041 v6.2.0]	"Technical realization of Cell Broadcast Service (CBS)", V6.2.0 (2003-12), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 6) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.041/
[3GPP 23.041 v7.0.0]	"Technical realization of Cell Broadcast Service (CBS)", V6.2.0 (2006-03), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 7) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.041/
[3GPP 23.038 v5.0.0]	"Alphabets and language-specific information", V5.0.0 (2002-03), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 5) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.038/
[3GPP 23.038 v6.0.0]	"Alphabets and language-specific information", V5.0.0 (2002-03), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 6) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.038/
[3GPP 23.038 v7.0.0]	"Alphabets and language-specific information", V5.0.0 (2002-03), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 7) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.038/
[3GPP 23.040 v5.0.0]	"Technical realization of the Short Message Service (SMS)", V5.0.0 (2001-06), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 5) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.040/
[3GPP 23.040 v6.0.0]	"Technical realization of the Short Message Service (SMS)", V5.0.0 (2002-09), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 6) URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.040/
[3GPP 23.040 v7.0.0]	"Technical realization of the Short Message Service (SMS)", V5.0.0 (2007-03), 3rd Generation Partnership Project; Technical Specification Group Terminals; (Release 7), URL:http://www.3gpp.org/ftp/Specs/archive/23_series/23.040/
[BCAST-TS-Service_ Guide]	"Service Guide for Mobile Broadcast Services", Version 1.0, Open Mobile Alliance™, OMA-TS-BCAST_Service_Guide-V1_0, URL:http://www.openmobilealliance.org/
[DCD-AD]	"OMA Dynamic Content Delivery Architecture", Version 1.0, Open Mobile Alliance™, OMA-AD-DCD-V1_0, URL:http://www.openmobilealliance.org/
[DCD-RD]	"OMA Dynamic Content Delivery Requirements", Version 1.0, Open Mobile Alliance™, OMA-RD-DCD-V1_0, URL:http://www.openmobilealliance.org/
[DCD-TS-BCAST]	"OMA Dynamic Content Delivery Technical Specification – BCAST Adaptation", Version 1.0, Open Mobile Alliance™, OMA-TS-DCD_Call_Flows-V1_0
[DCD-TS-CBS]	"OMA Dynamic Content Delivery Technical Specification – CBS Adaptation", Version 1.0, Open Mobile Alliance™, OMA-TS-DCD_CBS_Adaptation-V1_0, URL:http://www.openmobilealliance.org/
[DCD-TS-Charging]	"OMA Dynamic Content Delivery Technical Specification – Charging ", Version 1.0, Open Mobile Alliance™, OMA-TS-DCD_Charging-V1_0, URL:http://www.openmobilealliance.org/
[DCD-TS-Syntax]	"OMA Dynamic Content Delivery Technical Specification – Syntax", Version 1.0, Open Mobile Alliance™, OMA-TS-DCD_Syntax-V1_0, URL: http://www.openmobilealliance.org/

[DCD-TS-Bindings] "OMA Dynamic Content Delivery Technical Specification – Bindings", Version 1.0, Open Mobile Alliance™, OMA-TS-DCD_Bindings-V1_0, URL:http://www.openmobilealliance.org/

[DCD-XSD] "OMA Dynamic Content Delivery XML Schema", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_DCD-V1_0, URL:http://www.openmobilealliance.org/

[IANACharset] IANA MIBEnum Character Set Registry,
URL: ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets

[OMA-CBCS] "OMA Categorization Based Content Screening Draft V 1.0", Open Mobile Alliance™, OMA-TS-CBCS-V1_0, URL: http://www.openmobilealliance.org

[OMA-UAPROF] "OMA User Agent Profile V2.0", Open Mobile Alliance™, OMA-TS-UAProf-V2_0, URL:http://www.openmobilealliance.org/

[PUSH-OTA] "Push Over The Air", Version 25-April-2001, Open Mobile Alliance™, WAP-235-PushOTA-20010425-a, URL:http://www.openmobilealliance.org/

[PUSH-PAP] "Push Access Protocol", Version 29-Apr-2001, Open Mobile Alliance™, WAP-247-PAP-20010429-a, URL:http://www.openmobilealliance.org/

[RFC1321] "The MD5 Message-Digest Algorithm". R. Rivest, April 1992. URL:
URL:http://www.ietf.org/rfc/rfc1321.txt

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997,
URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC 2327] "SDP: Session Description Protocol", M. Handley, V. Jacobson, April 1998, URL:
URL:http://www.ietf.org/rfc/rfc2327.txt

[RFC2445] "Internet Calendaring and Scheduling Core Object Specification (iCalendar)". F. Dawson, D. Stenerson, November 1998, URL:http://www.ietf.org/rfc/rfc2445.txt

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1". Fielding R.; Gettys J.; Mogul J.; Frystyk H.; Masinter L.; Leach P.; Berners-Lee T., June 1999. URL:
URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC2617] "HTTP Authentication – Basic and Digest Access Authentication". J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, June 1999. URL:
URL:http://www.ietf.org/rfc/rfc2617.txt

[RFC 3266] "Support for IPv6 in Session Description Protocol (SDP)", S. Olson, G. Camarillo, A.B. Roach, June 2002, URL:http://www.ietf.org/rfc/rfc3266.txt

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[RFC4346] "The Transport Layer Security (TLS) Protocol Version 1.1", T. Dierks, E. Rescorla. April 2006, URL:http://www.ietf.org/rfc/rfc4346.txt

[RFC4347] "Datagram Transport Layer Security", E. Rescorla, N. Modadugu. April 2006,
URL:http://www.ietf.org/rfc/rfc4347.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[WBXML] "Binary XML Content Format Specification". WAP Forum™. WAP-192-WBXML.
URL:http://www.openmobilealliance.org/

2.2 Informative References

[OMA-DICT] "OMA Dictionary", Open Mobile Alliance™, OMA-Dictionary-V2_4, URL:http://www.openmobilealliance.org/

[SEC_CF] "OMA Security Common Functions", Open Mobile Alliance™, OMA-SEC_CF-V1.0 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

Application Deregistration	See [DCD-RD].
Application Profile	See [DCD-AD].
Application Registration	See [DCD-RD].
ATOM	See [DCD-AD].
Authentication	See [DCD-RD].
Cell Broadcast Service	A one-to-many geographically focused messaging service based upon the GSM/UMTS Short Message Service.
Channel Deregistration	The process of terminating availability of a DCD Channel.
Channel Discovery	The process of delivering Channel Guide information, to enable DCD Enabled Client Application awareness of available services, and to facilitate the Channel Subscription process.
Channel Guide	See [DCD-RD].
Channel Metadata	See [DCD-AD].
Channel Metadata Update	The process of delivering changes to Channel Metadata items for a specific DCD Channel, to applicable DCD Clients and DCD Enabled Client Applications.
Channel Registration	See [DCD-AD].
Channel Resumption	See [DCD-AD].
Channel Subscription	See [DCD-AD].
Channel Subscription Update	The process whereby a DCD Enabled Client Application updates its preferences related to a DCD Channel.
Channel Suspension	See [DCD-AD].
Channel Unsubscription	See [DCD-AD].
Client Activation	See [DCD-AD].
Client Deactivation	See [DCD-AD].
Content Metadata	See [DCD-AD].
Content Publication	See [DCD-AD].
Content Submission	The process of sending content from a DCD Enabled Client Application to a DCD Content Provider for arbitrary application purposes.
Content Update	See [DCD-RD].
DCD Channel	See [DCD-RD].Also referred to as “Channel”.
DCD Client	See [DCD-RD].
DCD Content Packaging	See [DCD-RD].
DCD Content Provider	See [DCD-RD].Also referred to as “Content Provider”.

DCD Enabled Client Application	See [DCD-RD].
DCD Server	See [DCD-RD].
DCD Service	See [DCD-RD].
DCD Service Provider	See [DCD-RD].Also referred to as “Service Provider”.
DCD XML	See [DCD-AD].
DCD-1	See [DCD-AD].
DCD-2	See [DCD-AD].
DCD-3	See [DCD-AD].
DCD-CADE	See [DCD-AD].
DCD-CAR	See [DCD-AD].
DCD-CPDE	See [DCD-AD].
DCD-CPR	See [DCD-AD].
Dynamic Content Delivery	See [DCD-AD].
HTTP Digest Authentication	A method of credentials negotiation between an HTTP client and HTTP server, defined in [RFC 2617].
Pull	See [DCD-AD].
Push	See [DCD-AD].
Push-OTA	See [DCD-AD].
Receive-Only Client Case	See [DCD-AD].
RSS	See [DCD-RD].
Subscription	See [DCD-RD].
Subscription Personalization	See [DCD-AD].
Subscription Validation	See [DCD-AD].
User Agent Profile	See [DCD-AD].
Wi-Fi	See [DCD-RD].

3.3 Abbreviations

AP	Application Profile
APN	Access Point Name
BCAST	The OMA BCAST enabler
CDF	Compound Document Formats
CBS	Cell Broadcast Service
CP	Content Provider
DCD	Dynamic Content Delivery
DTLS	Datagram Transport Layer Security
HTTP	Hypertext Transport Protocol
ID	Identifier
IP	Internet Protocol

MCC	Mobile Charging and Commerce
MIME	Multipurpose Internet Mail Extensions
MNO	Mobile Network Operator
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority, the operational naming authority established by WAP (as WAP Interim Naming Authority - WINA) and inherited by OMA.
PAP	Push Access Protocol
PLMN	Public Land Mobile Network
RADIUS	Remote Authentication Dial-In User Service (RFC 2865)
RFC	Request for Comments
RSS	Really Simple Syndication
TLS	Transport Layer Security
UAProf	User Agent Profile
UI	User Interface
UMTS	Universal Mobile Telecommunications System
URI	Universal Resource Identifier
URL	Universal Resource Locator
WAP	Wireless Application Protocol
WBXML	WAP Binary XML
WSP	Wireless Session Protocol
XDMS	XML Document Management Service

4. Introduction

This document starts with a description of the fundamental communication concept in DCD in this Section. Then, in Section 5, it continues by describing the lifecycle of transactions in DCD. Afterwards, in Section 6, the normative behaviours of operations are defined, followed by the transaction messages and the information elements in Section 7. Metadata elements and packaging schemas are each defined in Sections 8 and 9. Aspects of DCD security are covered in Section 10 while adaptations of DCD to transport bearers are defined for broadcast in Section 11 and point-to-point in Section 12. Further details for specific adaptation of DCD for BCAST and CBS should be referred to [DCD-TS-BCAST] and [DCD-TS-CBS] respectively. Error handling of all DCD operations and transactions are defined in Section 13. Section 14 defines the terminal capability disclosure method via UAProf. Section 15 defines a compact encoding for DCD, based upon WBXML. Section 16 defines how DCD Connection Profiles are managed using OMA Device Management, and how changes to Connection Profiles are handled.

The description and definitions in this document conform to the architectural model and protocol stack described in [DCD-AD].

4.1 DCD Deployment Model

DCD should support various network technologies, such as point-to-point and broadcast network technologies. Based upon each network technology or bearer used, the DCD enabler may use push or pull based mechanisms to facilitate the delivery of content over such bearers. The DCD enabler is designed to work on a point-to-point deployment scenario or broadcast scenarios [per the adaptation specifications] or combined.

4.2 Channel State Model

The main purpose of the DCD enabler is the delivery of DCD Content to the DCD Enabled Client Applications. The DCD Channel is defined as a primary delivery construct, therefore the state model of the DCD enabler is channel-centric. While the state of the channel is driven by the actions of external actors such as the DCD Enabled Client Application and the DCD Content Provider, it is the DCD Client and the DCD Server that implement the state changes. These actions define the lifecycle of the DCD enabler.

As far as content delivery is concerned, a delivery channel can be reduced to a simple “pipe” involving a single DCD Enabled Client Application, DCD Client, DCD Server, and a single Content Provider. Any other topologies (e.g. an application consuming content from multiple content providers) could be viewed as a composition of delivery channels. From the content delivery perspective, the states of the DCD Client and the DCD Server are correlated with the states of the DCD Channel, and could be amalgamated within the channel states (e.g. the DCD Channel may only be in an *Active* state if the DCD Client is “activated”).

The DCD Channel has the following states:

- ***Inactive*** – this is the initial state of the DCD Channel. The DCD Channel is in this state prior to registration of both the DCD Enabled Client Application and the DCD Content Provider, or after the DCD Enabled Client Application or the DCD Content Provider deregistered with the DCD Server. There is no content delivery in this state.

- ***Active*** – this state is associated with the delivery of the DCD Content. When the DCD Channel is in the *Active* state, the DCD Content Provider may deliver content to the DCD Enabled Client Application. Whether or not the content is being delivered (e.g. “dormant” DCD Channel) does not affect the state of the DCD Channel.

- ***Suspended*** – this state is a result of content delivery suspension request triggered by the DCD Enabled Client Application, by the DCD Service Provider, or by the DCD Content Provider. The suspension request is either explicit (i.e. using DCD-CADE and DCD-CPDE interfaces) or implicit (i.e. using Channel Metadata provided at registration).

The diagram in Figure 1 shows DCD Channel state transitions.

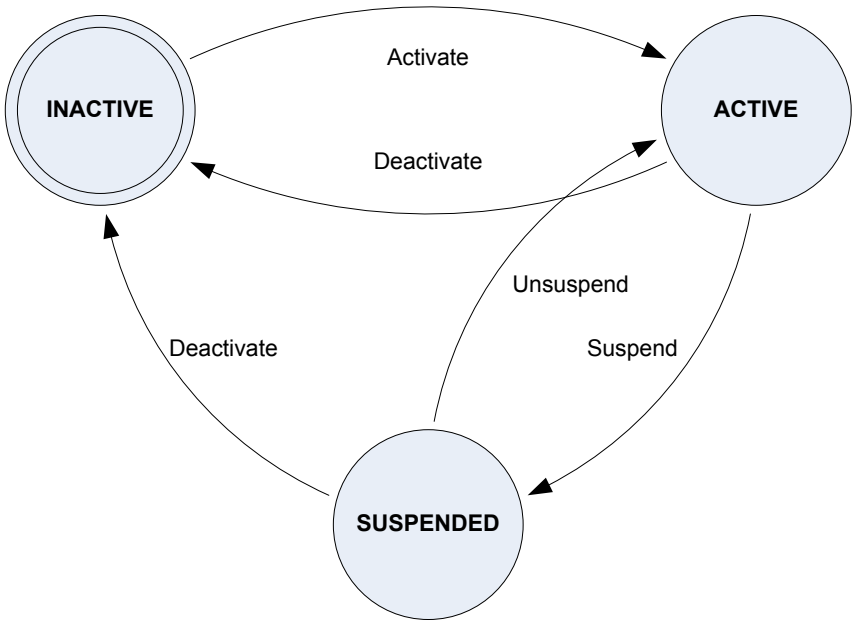


Figure 1: DCD Channel state transitions

Activate: The DCD Channel transition from the state *Inactive* to *Active* is a result of registration of both the DCD Enabled Client Application and the DCD Content Provider, and the DCD Enabled Client Application subscription to the DCD Channel. If the DCD enabler cannot bind the DCD Enabled Client Application with the Content Provider, (i.e. unable to match application preferences with registered Content Providers) the DCD Channel remains in the *Inactive* state. The prerequisite to establishing content delivery is successful registration of the DCD Enabled Client Application and of the matching DCD Content Provider. There is no requirement on the order in which these entities register with the DCD enabler.

Deactivate: The DCD Channel transition back to the *Inactive* state occurs upon deregistration of either the DCD Enabled Client Application or the DCD Content Provider. Deregistration is a result of one of the following events:

- the content channel is no longer available (i.e. deregistered)
- the channel metadata has changed and doesn't match application preferences anymore
- the application unsubscribed from the DCD Channel
- the application is uninstalled from the device and deregistered from the DCD Client

Suspend: The DCD Channel transition to the *Suspended* state occurs when the DCD Enabled Client Application, the DCD Service Provider, or the DCD Content Provider requests that the DCD enabler suspend content delivery. The content delivery suspension request may be issued directly using the DCD-CADE and DCD-CPDE interfaces or indirectly using settings in the Channel Metadata (e.g. suspend delivery when roaming).

Unsuspend: The DCD Channel transition from the *Suspended* to *Active* state occurs when the DCD Enabled Client Application, the DCD Service Provider, or the DCD Content Provider issues a request to resume channel delivery. The channel delivery may be resumed as a result of an interface action via DCD-CADE and DCD-CPDE or when the content delivery suspension condition specified in the Channel Metadata is no longer valid.

4.2.1 Receive-Only Client Case

In ***the receive-only case*** the defined state should be modified to the following definitions:

The DCD Channel has the following states:

- ***Inactive*** – this is the initial state of the DCD Channel. There is no content delivery in this state. The DCD Client is not "listening" to the receive-only channel or the Service provider has not yet defined the receive-only channel in the network.
- ***Active*** – this state is associated with the delivery of the DCD Content. When the DCD Channel is in the *Active* state, the DCD Content Provider may deliver content to the DCD Enabled Client Application. Whether or not the content is being delivered (e.g. "dormant" DCD Channel) does not affect the state of the DCD Channel. The DCD Client has to "listen" to the subscribed receive-only channel in this state.
- ***Suspended*** – this state is a result of content delivery suspension request triggered by the DCD Enabled Client Application. The suspension request is either explicit (i.e. using DCD-CADE interface) or implicit (i.e. using channel metadata settings, such as, suspend when roaming).

4.3 Implementation and Deployment Policies

Various aspects of DCD are subject to implementation and deployment decisions of the various DCD stakeholders, e.g. vendors (DCD Server, DCD Client, and terminals), DCD Service Providers, Content Providers, and DCD-Enabled Client Application developers. In the interest of balancing interoperability with implementation/deployment flexibility, DCD provides a limited set of options that can be selected by the various stakeholders. The options are provided in the following ways:

- Implementation options, i.e. a DCD functional aspect is optional for implementation. These are shown as optional in the SCR tables. Where optionality of a feature may impact interoperability, methods are described for disclosure or discovery of what functions are not implemented in other DCD entities, so that interoperability issues can be avoided.
- Deployment options, i.e. choice on selection of which DCD functions to use in particular service/delivery contexts. These options cover all cases in which a specific entity behaviour for a DCD function is not mandated. However, in some cases this "optional use" is described in the text, for clarity.
- Policy control for function use, i.e. constraints on the functions that entities are allowed to use, by other entities that are the policy controllers. The roles of the entities for these aspects are clarified in the text, through normative requirements. The choice of policy to use, and the effect of policy controls on entities, are implementation / deployment decisions of the various stakeholders.

For example:

- A DCD Service Provider may choose to offer only specific bearers, delivery methods, or personalization options to users' DCD-Enabled Client Applications, based upon various criteria, e.g. the basic data service plan to which the user is subscribed. Through the DCD-Enabled Client Application, the user may in turn choose to accept those terms of service, or may seek services from a different DCD Service Provider.
- A terminal vendor and supplier of an embedded DCD Client may choose to implement synchronous requests by DCD-Enabled Client Applications to the DCD Client, but not to support asynchronous content notification to the DCD-Enabled Client Applications ("Content Delivery without prior Content Request" as described in Section 7.3.2.2).

Policies can be expressed in various ways, e.g.:

- Through the protocol, by the parameters that entities choose, including in messages and channel/content metadata
- Through the device/user-agent characteristics in User Agent Profiles
- By the behaviour of the entities per configuration options provided by the implementers
- By the published policies of the service providers and vendors, e.g. through product information, developer guidelines, and service options/terms/conditions

Examples of specific implementation and deployment options, drawn from the normative sections below:

- DCD Service Provider deployment policy for connection security requirements of DCD-3 operations [Section 5.4]
- DCD Service Provider or DCD Content Provider deployment policy to support on-demand channel registration requests upon subscription requests for unregistered channels [Section 5.5]
- DCD Service Provider or DCD Content Provider deployment policy for scope of Channel Metadata parameters available for subscription personalization [Section 5.5.2], and disclosure of these through channel metadata
- DCD Service Provider deployment policy for applicable content delivery methods [Section 5.6], and disclosure of these through channel metadata
- Variations in DCD Client support for interactions with DCD Enabled Client Application, based upon the device execution environment, deployment policy, etc [Section 7.3]
- DCD Service Provider or DCD Content Provider deployment policy for use/values of various DCD metadata attributes, e.g. Application Profile, Channel Metadata, and Content Metadata [Section 8]
- DCD Service Provider and DCD Client deployment policy for authentication methods used in session establishment [Section 10.1], and disclosure of the supported/required methods during session establishment
- DCD Service Provider or DCD Content Provider deployment policy for connection security [Section 10.2.1], and disclosure of this through Connection Profiles for DCD interfaces
- Terminal / DCD Client vendors may select various DCD functions to support, and disclose the support through the UAProf published by the terminal vendor [Section 14]

5. DCD Lifecycle

This Section describes each step of the DCD lifecycle as defined the DCD AD. The following descriptions and diagrams are applicable for devices capable of bi-directional communication unless it is explicitly stated for ***receive-only clients*** (e.g. DVB-H, CBS, MBMS etc.).

5.1 Client Activation and Deactivation

The DCD Client activation is the prerequisite to the operations of DCD service, such as application registration with the DCD Server, content delivery, administrative operations, etc. The activation results in establishing the session between the DCD Client and the DCD Server (see Section 5.5). Subscriber authentication may be required in order to proceed with the activation and establish the session (see Sections 10.1 and 10.2).

The DCD Client activation process may be triggered by device provisioning, installation of the DCD Client, registration of the first DCD-Enabled Client Application (see Section 5.2), user's interaction, etc. Alternatively, the DCD Client activation may be triggered by a DCD Server; the DCD Server may request DCD Client activation on a single device or on multiple devices (e.g. new service activation via push or broadcast, controlled DCD Service deployment, etc.).

A single DCD Client may communicate with one or more DCD Servers resulting in one or more activation processes, each of these is performed with a separate DCD Server. The DCD Server address may be preconfigured in the DCD Client, supplied by an application or a DCD Server as a part of the DCD-3 connection profile, or provided by other means (e.g. OMA DM). The DCD Server address may be updated by a DCD Server as a part of the DCD-3 connection profile, or other means (e.g. OMA DM).

Upon activation, the DCD Client and the DCD Server establish the session. The session provides the context for registering the new DCD-Enabled Client Applications, setting up the delivery options as part of the Application Profile and Channel Metadata, performing content delivery, administrative actions, etc.

The details of client activation are defined in Section 7.1.3.1.

Deactivation of DCD Client occurs upon the deactivation request from the DCD Client (see Section 7.1.3.2) or following the deactivation notification from the DCD Server (see Section 7.1.3.2.2). Deactivation may be a result of deregistration of all DCD Enabled Client Applications, cancellation or suspension of the DCD service, etc. Deactivation results in session termination and means tearing the association between the DCD Client and the DCD Server. Deactivation doesn't imply termination of the DCD Client, moreover, upon deactivation with one DCD Server the DCD Client may have valid sessions with other DCD Servers.

Figure 2: Client Activation options: (1) unauthenticated activation, (2) authenticated activation, (3) unauthenticated activation by server request, (4) authenticated activation by server request

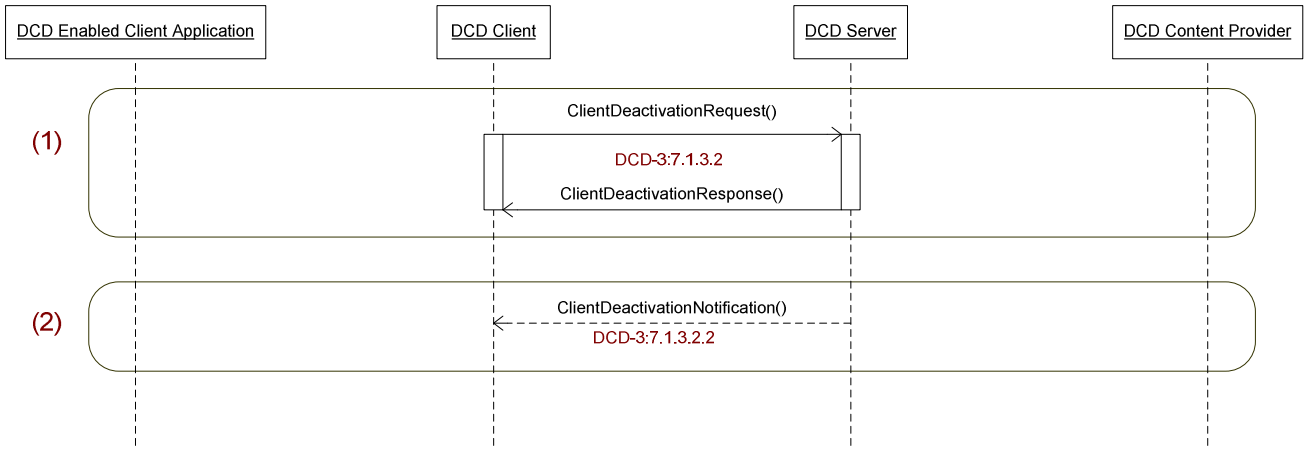


Figure 3: Client Deactivation options: (1) DCD Client requesting deactivation (2) DCD Server requesting deactivation

5.1.1 Receive-Only Client Case

In ***receive-only clients***, client activation means that the DCD Client begins listening to the DCD-3 channel.

The DCD Client activation process may be triggered by device provisioning, installation of the DCD Client, registration of the first DCD-Enabled Client Application (see Section 5.2), user interaction, etc.

The DCD-3 connection profile that the DCD Client should use to listen to the DCD Server will be established through device provisioning or by Application Profile (DCD-3 connection profile) reception in the DCD Client as part of application registration process.

In the broadcast only case session-id is not allocated during the Client Activation process and therefore it is not used during the service lifecycle.

Deactivation occurs when the DCD Client stops listening to the DCD-3 channel. It may be triggered as a result of deregistration of the last DCD Enabled Client Application or by explicit command from the DCD Server.

5.2 Registration

5.2.1 Application Registration

The DCD Enabled Client Application needs to be registered with the DCD enabler in order to receive DCD Content. An application registers with the DCD enabler using the DCD-CAR interface of the DCD Client. Some general purpose applications (e.g. browser, RSS viewer, etc.) could be registered indirectly (i.e. by another application, device runtime, DCD Client, etc.) At registration, the DCD Client receives an Application Profile, which contains Channel Metadata for all channels supported by the application along with the application settings common for all channels (e.g. application-ID, DCD-3 connection profile, etc.).

Following application registration with the DCD Client, the DCD Client registers the application with the DCD Server. The registration message to the DCD Server contains an entire Application Profile or a subset of the Application Profile. The DCD Server matches the DCD Enabled Client Application with one or more registered content channels based on the preferences in the Application Profile. These preferences could be explicit (i.e. a channel-ID) or implicit (i.e. the application-ID or the content types supported by the application). The registration response from the DCD Server contains the "Channel Guide" which is the list of Channel Metadata for the channels matching application preferences. The registration process may be followed by a subscription to the channels offered on the Channel Guide.

Channel Metadata within the Application Profile submitted by the DCD Enabled Client Application at registration could be changed during the subscription process or following the subscription (e.g. when device capabilities have changed).

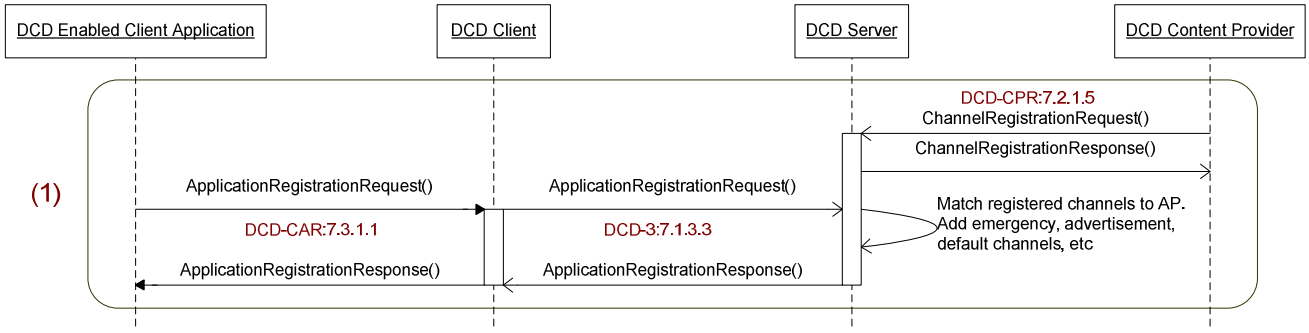


Figure 4: Application Registration

5.2.1.1 Receive-Only Client Case

In ***receive-only clients***, as a consequence of the registration process, the DCD Client matches the DCD Enabled Client Application with one or more content channels, discovered through the Channel Discovery process based on the preferences in the Application Profile. These preferences could be explicit (i.e. a channel-ID) or implicit (i.e. an application-ID or the content types supported by the application).

5.2.2 Channel Registration

The DCD Content Provider registers with the DCD enabler by interacting with the DCD-CPR interface of the DCD Server. The DCD-CPR interface supports dynamic channel registration and allows Content Providers to "plug-in" the new channels into the DCD enabler. At registration, the Content Provider supplies Channel Metadata for offered DCD Channels. This metadata contains settings used by the DCD Server and the DCD Client to establish content delivery to appropriate DCD Enabled Client Applications. Similarly to integrating general purpose device applications with the DCD enabler, a Content Provider may be engaged with the DCD enabler through indirect, mediated registration performed by a DCD Service Provider or any other authorized entity.

The DCD enabler matches registered DCD Enabled Client Applications and DCD Content Providers. Upon the registration, the DCD Enabled Client Applications receive a list of matching channels in the registration response. Upon the registration of the new channel, the DCD enabler may send channel availability notifications to the devices with matching applications installed. The binding between a DCD Enabled Client Application and content channels is dynamic and may change according to the settings defined by the Application Profile. The DCD enabler uses explicit or implicit binding to match the applications and the Content Providers. In the case of explicit binding, the DCD enabler matches an application with a Content Provider using a channel-ID supplied in the Application Profile. Alternatively, the DCD enabler may match an application with a Content Provider using an application-ID specified in the Channel Metadata provided by a Content Provider. In the case of implicit binding, the DCD enabler matches the content types supported by the DCD-Enabled Client Application, as described by its Application Profile, with the content types for registered DCD Channels.

DCD Content Provider may use registration process through DCD-CPR interface to change its channel offering at any time. These changes may result in channel availability notifications being sent to the devices with appropriate DCD Enabled Client Applications installed. DCD Content Provider initiated changes in Channel Metadata may trigger application unsubscription from these channels and/or termination of content delivery.

The details of application and content provider registration messages and of channel availability notification message are defined in 7.1.3.3 , 7.1.3.10, and 7.2.1.5.

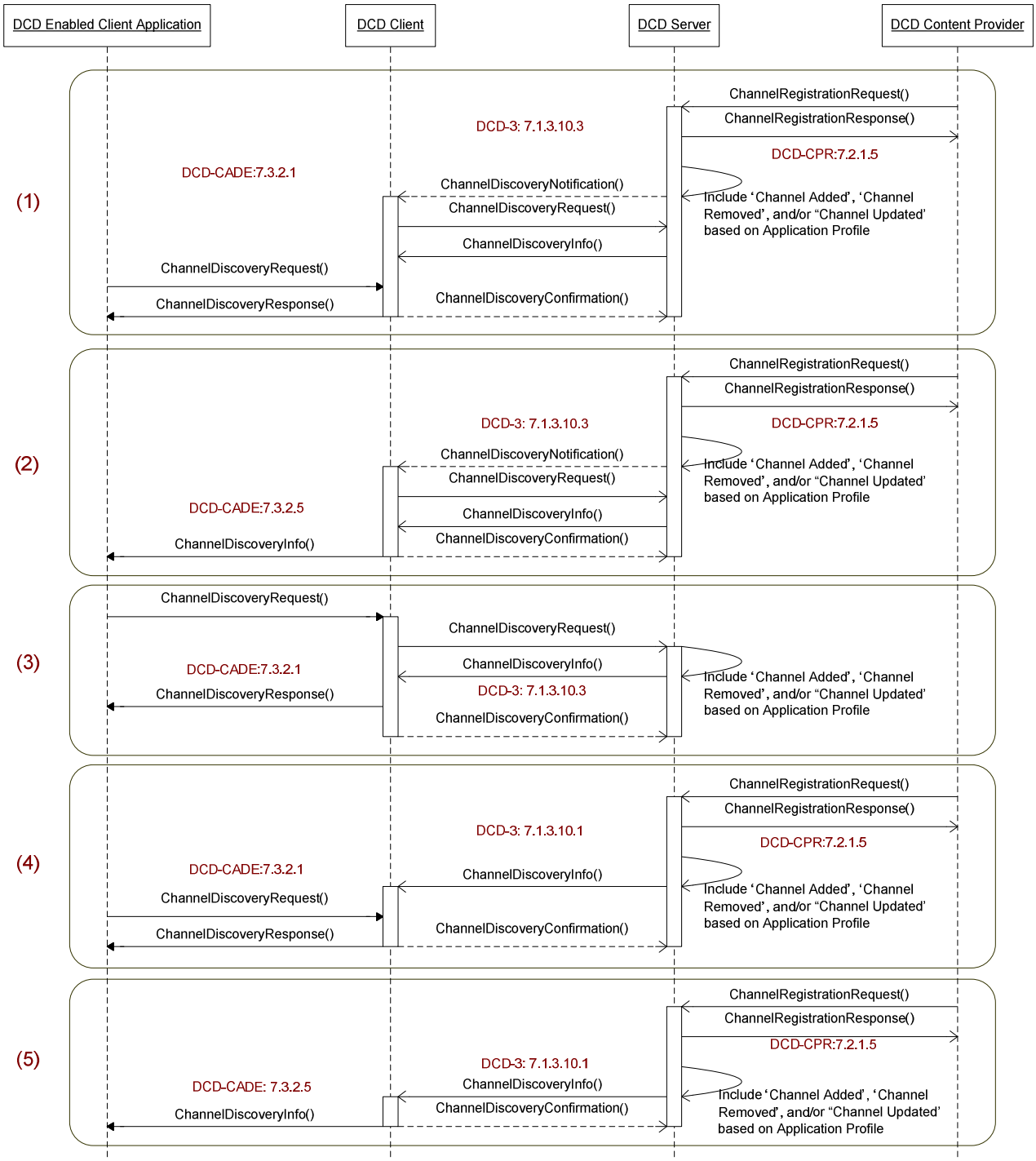


Figure 5: Channel Registration options: (1) channel discovery notification and subsequent request from the application, (2) channel discovery notification with push of channel discovery info to the application, (3) channel discovery initiated by the application, (4) channel discovery information push and subsequent request from the application, (5) channel discovery information push to the application

5.2.3 Channel Metadata Update

The DCD Content Provider updates the channel metadata by interacting with the DCD-CPR interface of the DCD Server. The DCD-CPR interface supports dynamic channel metadata update. Upon the update of channel metadata for the subscribed channel, the DCD Server notifies the modified channel metadata to the DCD Client and/or subscribed DCD Enabled Client Application. The channel metadata update may also be originated by the DCD Server.

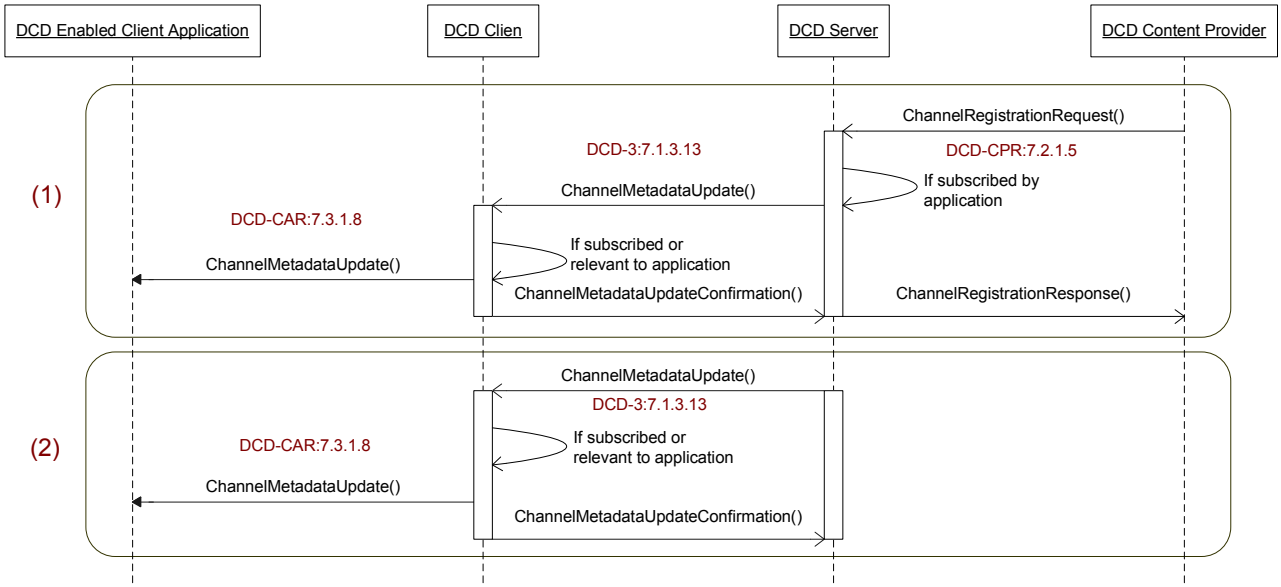


Figure 6: Channel Metadata Update: (1) Update the channel metadata from the Content Provider affecting the DCD Client or DCD Enabled Client Application, (2) Channel Metadata Update originated by the DCD Server

5.3 Deregistration

5.3.1 Application Deregistration

The DCD Enabled Client Application deregisters with the DCD enabler by sending a request to the DCD Client over DCD-CAR interface (see Section 7.3.1.2 for details). A deregistration request could be issued as a result of user action, software removal, etc. Alternatively, the DCD Client may deregister an application without receiving deregistration request (e.g. when it discovers that the application is no longer available). The DCD Client notifies the DCD Server about application deregistration providing the application-ID of the application (see Section 7.1.3.4 for details). The DCD Enabled Client Application should unsubscribe from all subscribed DCD Channels prior to deregistration. If application deregistration is initiated by the DCD Client, the DCD Client needs to unsubscribe from all outstanding subscriptions on behalf of the application.

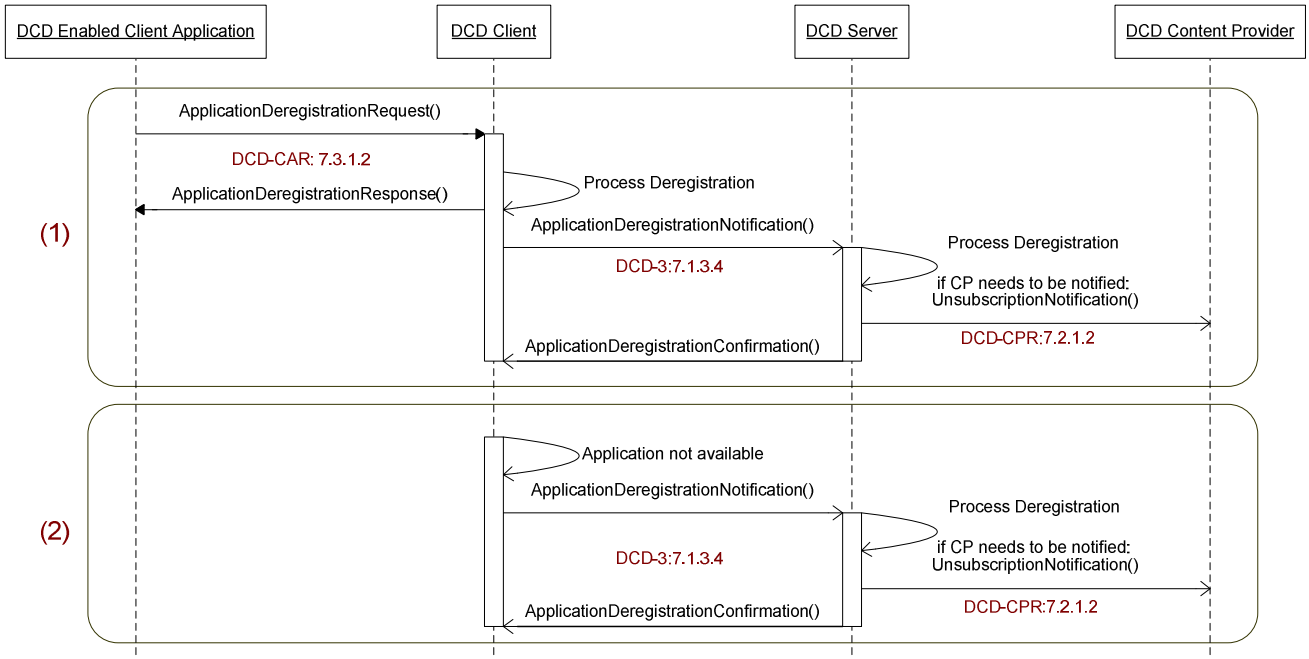


Figure 7: Application Deregistration: (1) the application deregistration, (2) the application deregistration initiated by software removal

5.3.1.1 Receive-Only Client Case

In ***receive-only clients***, when the last DCD Enabled Client Application is deregistered from the DCD Client, the DCD Client is expected to stop listening to the DCD-3 broadcast channel (Deactivation).

5.3.2 Channel Deregistration

The DCD Content Provider deregisters channels with the DCD enabler by interacting with the DCD-CPR interface of the DCD Server. At deregistration, the Content Provider supplies channel-ID of DCD Channels that are deregistered. The channel deregistration process could initiate the channel unsubscription notification to the DCD Client and to the DCD Enabled Client Application. The channel deregistration process could initiate the channel discovery transaction that lists the channel as removed. Such channel discovery transactions may occur immediately or later per DCD Service Provider policy.

Alternatively, deregistration of the Content Provider channels may also be done in the DCD Server without request from the Content Provider. The DCD Server could send channel unsubscription notification to the DCD Clients subscribed to the channel being deregistered. The DCD Client in turn could notify the DCD Enabled Client Application of the unsubscription. DCD Server may also notify the Content Provider about deregistration of the channel.

DCD Content Provider may use deregistration process through DCD-CPR interface to change its channel offering at any time.

The details of Content Provider Channel Deregistration messages are defined in Section 7.2.1.6.

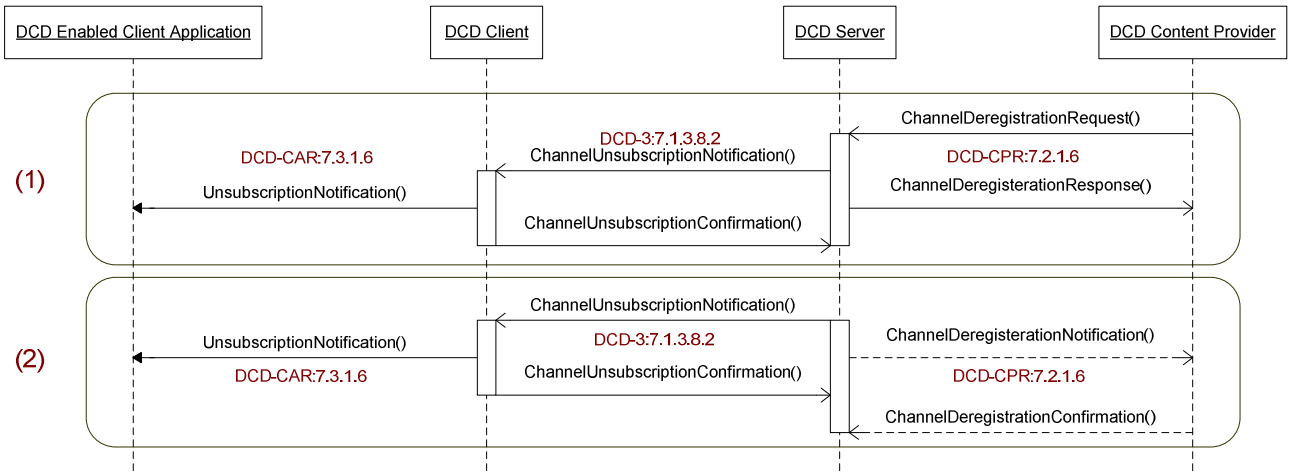


Figure 8: Channel Deregistration options: (1) requested by the Content Provider, (2) requested by the DCD Server

5.4 Session Management

A session is a framework within which the DCD Client and the DCD Server communicate over the DCD interfaces for the purposes of providing a DCD service, such as the negotiation of DCD version and device capabilities, establishment of content delivery, and suspension / resumption of the DCD Channels. The term “session” in this context is not a transport-layer session, but rather an application session. It is therefore transport-independent. If the transport connection is broken or the device is power-cycled, the client can reconnect the transport connection and, if successful, it will be able to continue the previously established session.

A session is established upon client activation (see Section 7.1.3.1). While establishing a session, the DCD Client and DCD Server authenticate themselves via the DCD-3 interface (see Section 10.1), and comply with the connection security requirements for DCD-3 operations per the policy of the Service Provider (see Section 10.2).

The DCD Client may initiate a session with the highest version of the protocol and the security level that it supports. The DCD Client may also re-establish a new session at any time during an active session. The DCD server will deny the session establishment if it does not support the requested DCD protocol version. If the DCD Client is denied session establishment the DCD Server SHALL respond providing the protocol version supported by the DCD Server and the DCD Client may retry session establishment. A session SHALL be instantiated per DCD Client.

Associated with each session is an auto expiry timer value called *Session Time to Live (TTL)* which is set by the DCD Server. The session becomes invalid when the *Session TTL* expires, or either the DCD Client or the DCD Server decides to end the session via client deactivation procedure (see Section 7.1.3.2).

A session is identified by a *Session-ID*, which is assigned upon session establishment. The *Session-ID* is unique within a DCD Server Domain. The *Session-ID* is associated with multiple Application-IDs, each of which refers to an Application Profile for a specific DCD Enabled Client Application and its capabilities expressed as metadata.

During a session, both the DCD Client and the DCD Server preserve the metadata describing the negotiated version and capabilities. The device is able to update the negotiated version and capabilities dynamically i.e. by re-activation if its capabilities have changed. The Session-ID however cannot be dynamically altered during the lifetime of the session.

5.4.1 Broadcast Service Identifier for DCD over CBS

In a CBS environment, the one-to-one “session” relationship between DCD Client and DCD Server is not usable for the same purpose as in point-to-point environments, e.g. for recognition/authentication of the other entity in DCD transactions. While a session may be active for DCD-1, DCD-2, and DCD-3 interface transactions across point-to-point bearers, the associated

session ID cannot be used for broadcast-delivered messages. In CBS, the same broadcast bearer may be used by multiple DCD Service Providers. To provide a means to distinguish DCD services, e.g. provided by one or more DCD Service Providers in such environments, DCD defines the Broadcast Service Identifier ("Broadcast Service ID").

The Broadcast Service ID is used as an identifier for the source of DCD content delivered over CBS bearers, and may represent a single source, e.g. a specific DCD Service Provider, or a collective source e.g. a common DCD service provided via multiple DCD Service Providers.

Broadcast Service IDs may be valid for a limited duration. The validity time of Broadcast Service IDs are set by the allocating authority, e.g. DCD Service Providers, for the Broadcast Service IDs specific to them.

In order to achieve uniqueness for DCD services as necessary, the Broadcast Service ID should start with a registered internet domain name. This enables uniqueness for DCD Service Provider specific services, e.g. based upon a domain registered to the DCD Service Provider, or for shared services based upon another unique domain. It also provides flexibility to the relationships between DCD Service Providers and Mobile Network Operators (MNO), e.g. the MNO may be the DCD Service Provider, or provide CBS service to 3rd party DCD Service Providers, who possibly also receive CBS service from other MNO's.

As applicable, Broadcast Service IDs are provided to DCD Clients at Activation, as part of the Application Profile during Application Registration or as a preset value using DCD MO or any other proprietary method. The presence of this parameter is conditional on the support of CBS in the DCD Server and Client. Further details on use of the Broadcast Service ID are provided in [DCD-TS-CBS].

5.5 Subscription

To start content reception, the DCD-Enabled Client Application needs to perform a subscription to the DCD Channel. The application may also be auto subscribed "out-of-the-box" (e.g. for emergency or other default channels). A subscription is an agreement between the DCD-Enabled Client Application and service provider that the DCD-Enabled Client Application is entitled to receive content for the specific DCD Channel.

In DCD a subscription can be established in two ways:

- *internally* in the enabler based on the channel discovery information provided to a DCD Enabled Client Application in the Channel Guide or via channel discovery transactions (see Section 7.1.3.10). Alternatively, the DCD Enabled Client Application may receive reference to a channel not yet registered with the DCD Server and to request subscription involving registration of such channel. The DCD Enabled Client Application may obtain channel-id or channel reference as a part of DCD Content (e.g. RSS feed) or by other proprietary means. *Internal subscription* is always initiated by the DCD Enabled Client Application sending subscription message to the DCD Client and completed by the confirmation from the DCD Server and/or the DCD Content Provider. Such subscription is called *internal* as it is started and completed using the entities of the DCD enabler.
- *externally* to the enabler (e.g. originated via web browser). *External subscription* is always initiated with the subscription portal (e.g. some website) located *outside* the DCD enabler and based on channel discovery information provided by this portal. The notification about subscription is received by a DCD Server and to complete the subscription the DCD Server needs to validate it with targeted DCD Enabled Client Application. Such subscription is called *external* as it is always started outside the DCD enabler and can be initiated by entities not belonging to the DCD enabler (e.g. web browser).

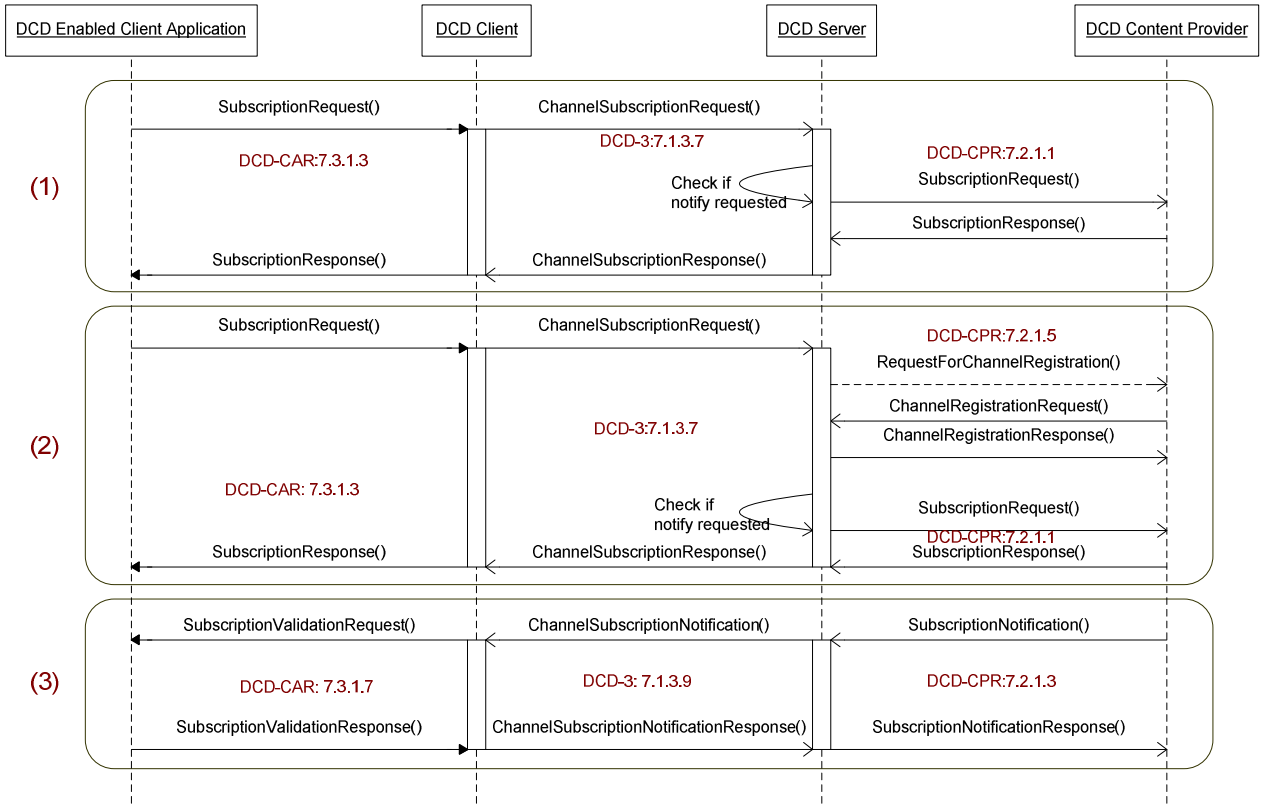


Figure 9: Subscription options: (1) internal subscription for registered channel, (2) internal subscription for unregistered channel (3) external subscription

5.5.1 Receive-Only Client Case

In ***receive-only clients***, subscription to a channel means that the DCD Client begins listening to the broadcast channel using the parameters specified in the DCD-2 connection profile and propagates data received to the DCD Enabled Client Application.

Also in ***receive-only clients***, a subscription can be established based on information received through channel discovery (see Section 7.1.3.10) or based on information obtained outside the enabler (e.g., application “knowledge” or user “knowledge”).

5.5.2 Subscription Personalization

Subscription personalization is fulfilled by the DCD enabler and by the DCD Content Provider. As the DCD Content is opaque for the DCD enabler, personalization by the DCD enabler only involves personalization options applicable to all channels (e.g. by location, presence, user / device profile, etc.). These options are expressed in Channel Metadata provided at subscription. With subscription personalization by the DCD enabler, the DCD Server performs adaptation of DCD Channel content in accordance with subscriber’s preferences expressed in Channel Metadata (see Section 8.2 for details). The DCD Enabled Client Application and the DCD Client can update personalization related Channel Metadata at any time during the subscription. The scope of Channel Metadata parameters for subscription personalization could be limited by policy of the DCD Service Provider or the DCD Content Provider.

Subscription personalization by the DCD Content Provider occurs outside the DCD enabler and typically associated with the type of content for a particular DCD Channel. Such personalization results in grouping users according to the subscription preferences (e.g. subscription filter). Such group is identified by the subscription identifier issued by the DCD Content

Provider and provided to the DCD enabler for content delivery purposes. When the DCD Content Provider specifies the subscription identifier along with the content, the enabler delivers such personalized content to a group of DCD Channel subscribers established according to this subscription identifier.

Both personalization types are applicable for the internal and external subscriptions. The subscription transactions shown in Figure 9 involve subscription personalization, as personalization information may be submitted as a part of Channel Metadata provided at subscription. Figure 10 shows additional transactions related to subscription personalization, both by the DCD enabler and by the DCD Content Provider.

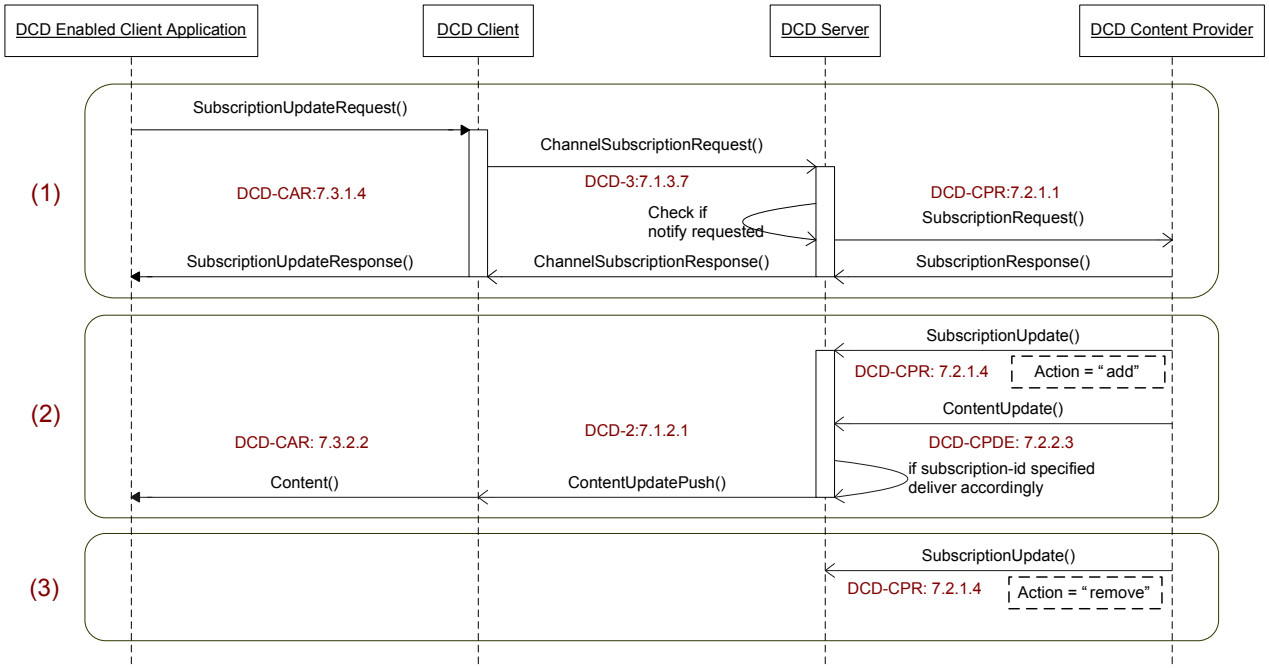


Figure 10: Subscription Personalization related transactions: (1) Personalization by the DCD enabler: update of subscription preferences by the application, (2) Personalization by the DCD Content Provider: adding subscriber to the group followed by content delivery to this group (according to subscription-id), (3) removing the subscriber from the group

5.5.2.1 Receive-Only Client Case

In ***receive-only clients***, with subscription personalization by the DCD enabler, the DCD Client has to perform adaptation of DCD Channel content in accordance with subscriber's preferences in Channel Metadata (see Section 8.2 for details). The DCD Enabled Client Application and the DCD Client may update personalization related Channel Metadata at any time during the subscription. The scope of Channel Metadata parameters for subscription personalization could be limited by policy of the DCD Service Provider or the DCD Content Provider.

5.5.3 Channel Unsubscription

A subscription becomes invalid as a result of an unsubscription procedure or when the end criteria of the subscription are reached (e.g. subscription period expiration).

Similarly to subscription, the unsubscription can also be performed internally or externally to DCD enabler. The flows of events in these approaches SHALL be analogous to subscription procedure (see Section 5.5 for subscription flows).

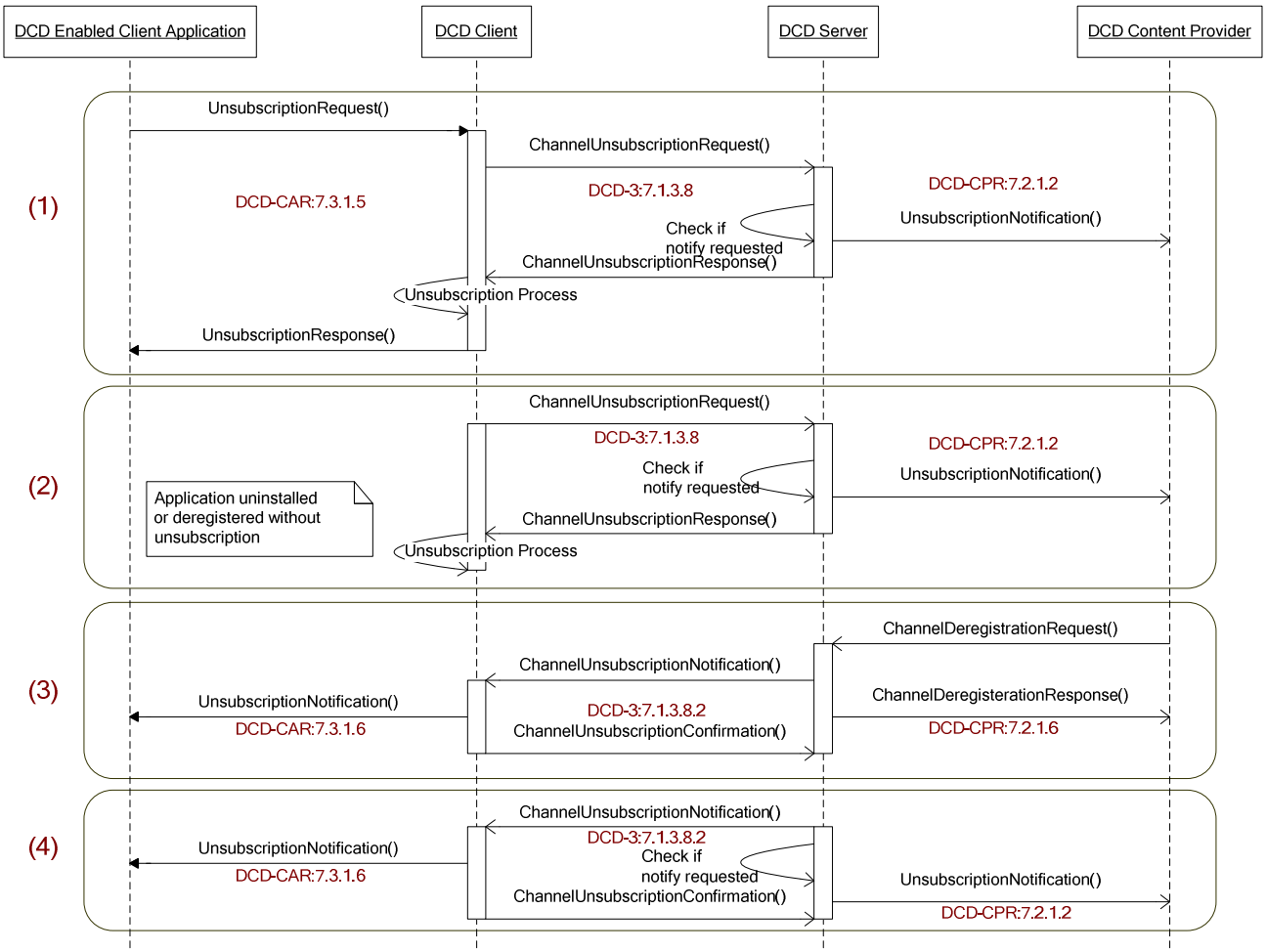


Figure 11: Channel Unsubscription options: (1) unsubscription from User, (2) Application uninstalled or deregistered without unsubscription, (3) deregistration from the Content Provider, (4) unsubscription from the DCD Server upon subscription expiration

5.5.3.1 Receive-Only Client Case

In ***receive-only clients***, a subscription becomes invalid as a result of an unsubscription procedure initiated from the DCD Enabled Client Application (i.e. terminating “listening”) or when decided by the Service provider (i.e. terminating “sending”).

5.6 Delivery

Content delivery occurs over DCD-1 or DCD-2 interfaces. The DCD Channel as a primary delivery construct carries DCD Content of interest to the DCD Enabled Client Application. The DCD Service Provider chooses the applicable content delivery methods based on the service policy and capabilities. The DCD Content could be delivered upon availability (i.e. upon retrieval from the DCD Content Provider or upon being published at the DCD Server), on schedule, or upon application request. In delivery upon availability mode, the push messages could carry the DCD Content or the notification of the DCD Content availability. In the delivery on schedule mode, the DCD Client can contact the DCD Server upon the predefined schedule. The delivery schedule may be established by the DCD Service Provider. A single DCD Channel may mix multiple delivery modes (e.g. Push, Pull). For example, the RSS content could be delivered upon availability, but the content from the links embedded in the RSS document could be delivered upon application request following content consumption by the user.

The main feature of the DCD enabler is the ability to manage delivery of generic content using metadata supplied by the applications and content providers. Channel establishment is managed by the Application Profile (see Section 8.1) provided by the DCD Enabled Client Application and by Channel Metadata (see Section 8.2) supplied by the DCD Content Provider. The DCD enabler uses Channel Metadata to establish general content handling settings for the DCD Channel. The DCD Content delivered over the DCD Channel is augmented by the Content Metadata (see Section 8.3). The Content Metadata contains delivery settings for a particular content item and overwrites associated general settings, if defined in Channel Metadata.

A DCD Channel is typically associated with a set of content types and MIME types. Based on these types and based on the attributes provided in the Application Profile / Channel Metadata, the DCD enabler may use different bearer networks for different DCD Channels. The attributes provided in Content Metadata may affect the choice of delivery bearer for the particular content item. This choice is based on delivery cost, bandwidth, quality of service (e.g. latency), etc. For example, the DCD enabler may deliver “movie channel” content as follows: movie guide over any available network, movie preview over 3G (if in “home” network) or over Wi-Fi, full movie over Wi-Fi only. The DCD Server makes the decision about the delivery network bearer based on device capabilities and dynamic information provided by the Contextual Information Upload (see Section 7.1.3.5) or the User Agent Profile provided by the DCD Client (see Section 12.2.1).

Content delivery over DCD Channel could be suspended upon the request from the DCD Enabled Client Application or from the DCD Content Provider (see Section 5.6.1), or terminated upon unsubscription (see Section 5.5.3) or / and deregistration of the DCD Enabled Client Application or the DCD Content Provider (see Section 5.3).

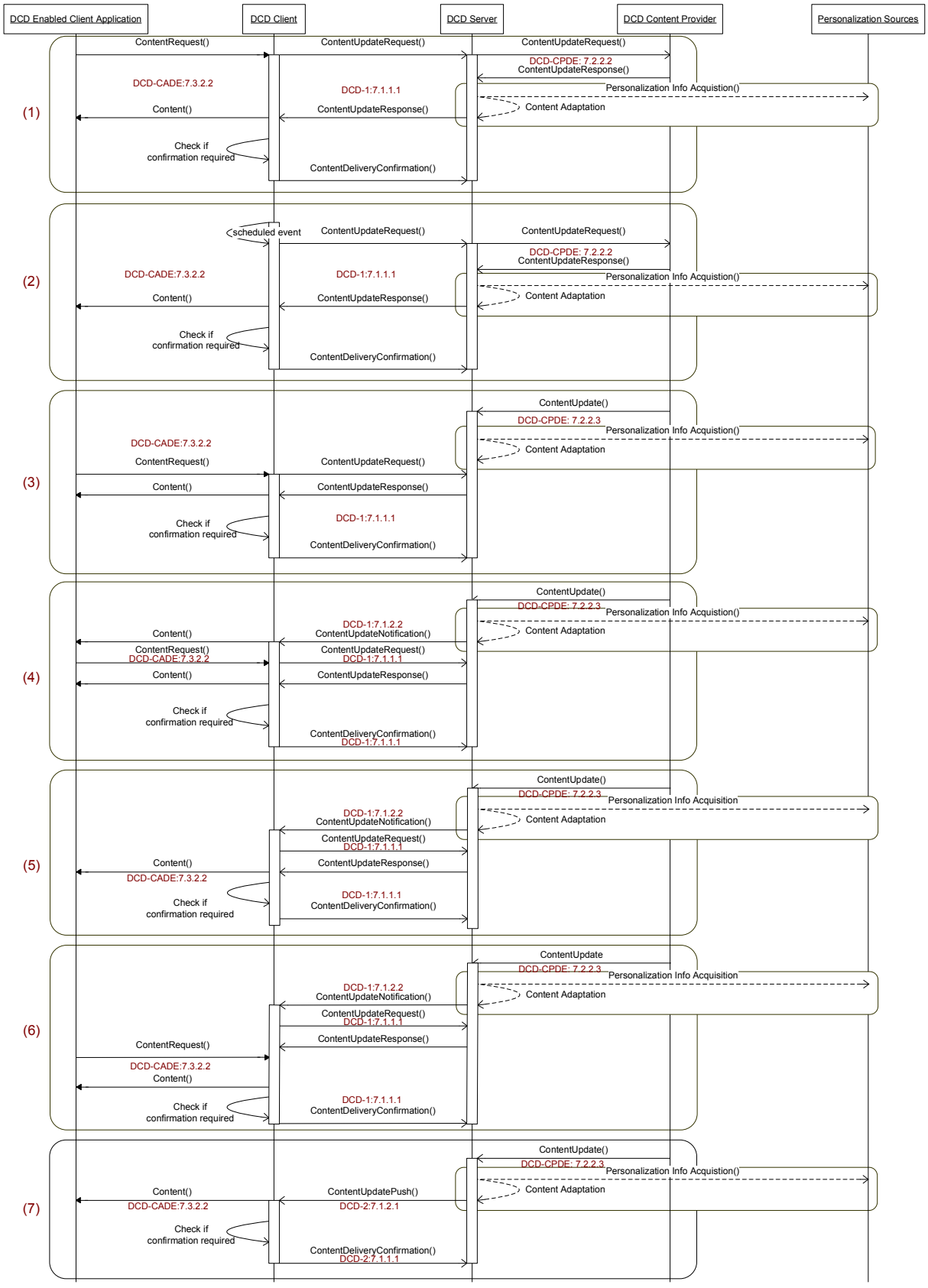


Figure 12: Delivery scenarios: (1) requested by application, (2) requested by DCD Client as per schedule in the channel metadata, (3) content published by Content Provider and requested by application, (4-6) content published by Content Provider with notification to DCD Client, (7) content published by Content Provider with content push to DCD Client and application

5.6.1 Receive-Only Client Case

In ***receive-only clients***, content delivery occurs over the DCD-2 interface. The DCD Channel as a primary delivery construct carries DCD Content of interest to the DCD Enabled Client Application. The DCD Content could be delivered upon availability (i.e. upon retrieval from the DCD Content Provider or upon being published at the DCD Server) or based on schedule.

The broadcasted messages transmitted over the DCD-2 interface carry the DCD Content (including content metadata).

The delivery schedule will be set in the DCD Server by the Service Provider.

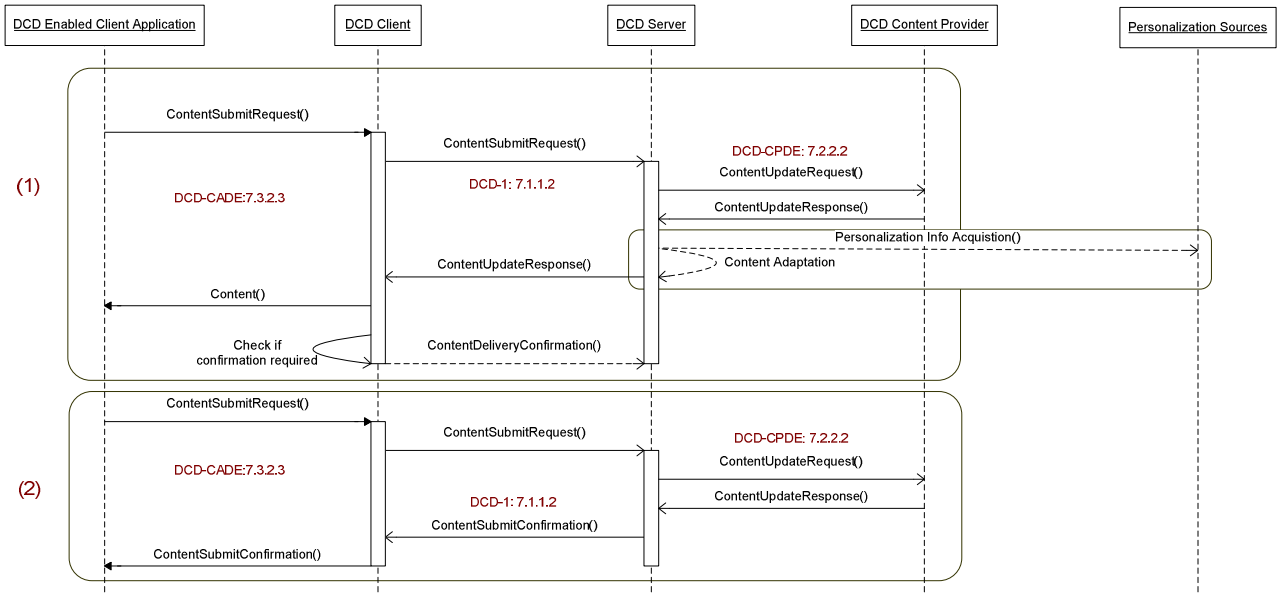
The DCD Server makes the decision about the delivery network bearer based on device capabilities and network capabilities

5.7 Content Submission

Content submission occurs over the DCD-1 interface. Content submission allows DCD Content Providers to create DCD-enabled services featuring interaction with the DCD-Enabled Client Application for various purposes, e.g.:

- Submitting user or application created content, e.g. for a DCD-enabled photo blogging/sharing service.
- Form submission, e.g. for a DCD-enabled content search service.
- Personalizing service through submission of preferences to the DCD Content Provider.
- Subscription operations with the DCD Content Provider.

The DCD enabler facilitates these transactions, but is uninvolved in the details or handling of the submitted content. In its response, the DCD Content Provider may include content to be delivered to the DCD Client.



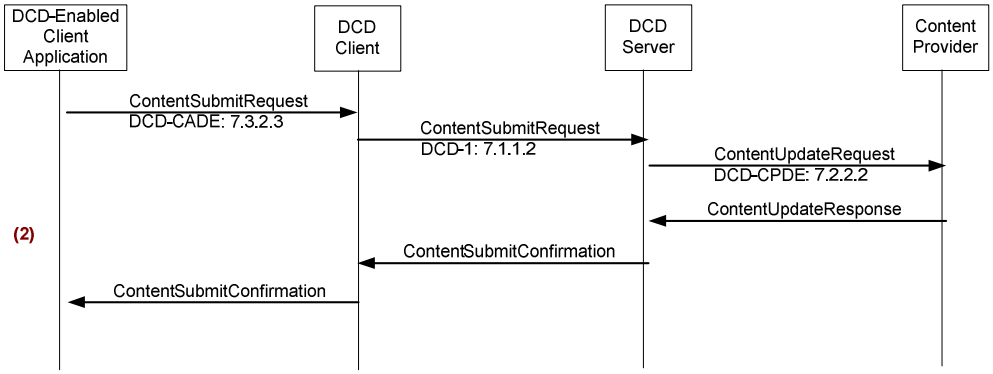


Figure 13: Content Submission: (1) with DCD Content returned, (2) without DCD Content returned

5.8 Suspension and Resumption

5.8.1 Channel Suspension

Channel Suspension can be initiated by a DCD Client or a DCD Server. The DCD Client issues Channel Suspend Request message (see Section 7.1.3.11.1) upon the request from the DCD Enabled Client Application (see Section 7.3.2.4.1) or when the suspension conditions in the Channel Metadata are satisfied (e.g. channel suspension when roaming). The DCD Enabled Client Application request to suspend a channel may not result in the DCD Client sending the suspension request to the DCD Server if the channel is shared by multiple applications on the device. The Channel Suspend Request is an outcome of user unwillingness to have content delivered to their device due to some temporary conditions (e.g. roaming, insufficient storage, etc.)

The DCD Server issues Channel Suspend Notification (see Section 7.1.3.11.5) upon:

- suspend notification from the DCD Content Provider (see Section 7.2.2.1.5);
- when the suspension conditions in the Channel Metadata are satisfied. These conditions could represent user preferences provided by the DCD Enabled Client Application or general channel preferences from the DCD Content Provider;
- when the DCD Service Provider chooses to suspend a DCD Channel for a single user or a group of users (e.g. due to non-payment) or for all users (e.g. due to the issues with channel content)

If channel suspension is requested by the DCD Content Provider it affects all users subscribed to this DCD Channel and such suspension is associated with some temporary conditions at the content provider.

Both the DCD Client and the DCD Server may issue suspension for one, several, or all DCD Channels.

5.8.2 Channel Resumption

Channel Resumption can be initiated by a DCD Client or a DCD Server. The DCD Client can only resume the channels previously suspended by the DCD Client and the DCD Server can only resume the channels previously suspended by the DCD Server. The DCD Client issues Channel Resume Request message (see Section 7.1.3.11.3) upon the request from the DCD Enabled Client Application (see Section 7.3.2.4.3) or when the suspension conditions in the Channel Metadata are no longer satisfied (e.g. user returned to home network).

The DCD Server issues Channel Resume Notification (see Section 7.1.3.11.7) upon:

- channel resume notification from the DCD Content Provider (see Section 7.2.2.1.7), if previously suspended due to the DCD Content Provider request;

- when the suspension conditions in the Channel Metadata is no longer satisfied, if previously suspended due to the suspension conditions in the Channel Metadata;
- when the DCD Service Provider chooses to resume the channel, if previously suspended by the Service Provider

Both, the DCD Client and the DCD Server may resume one, several, or all suspended DCD Channels.

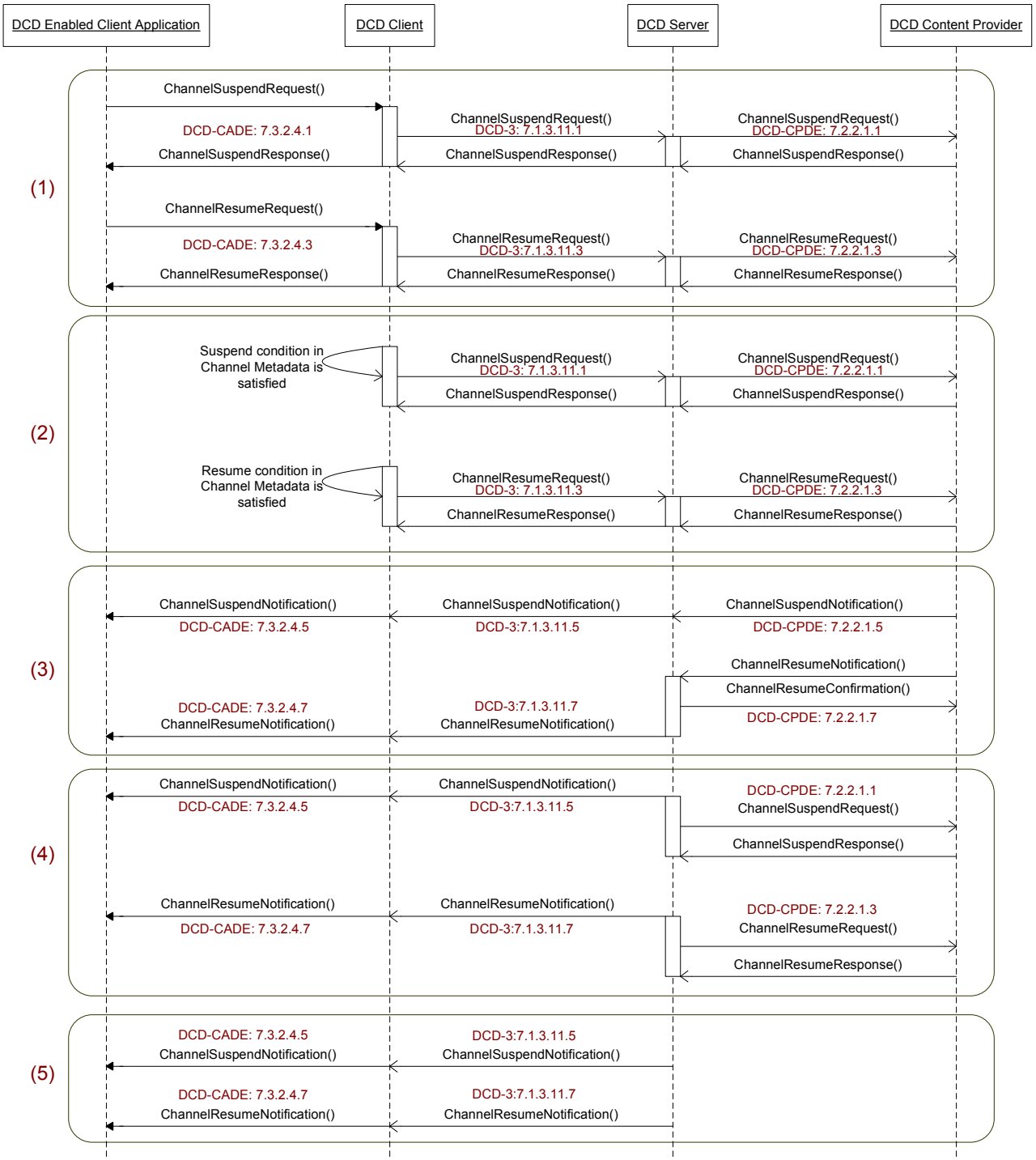


Figure 14: Suspend and Resume options: (1) requested by user, (2) based on the channel metadata condition, (3) requested by Content Provider, (4) requested by DCD Server and notification to the Content Provider, (5) requested by DCD Server

5.9 Operation Interface Mapping

	CAR/CADE		DCD-1/DCD-2/DCD-3					CPR/CPDE	
Operation	**Pull**	**Push**	**Pull**	**Push**		**Notification**		**Pull**	**Push**
			PTP	**PTP**	**Broadcast**	**PTP**	**Broadcast**		
DCD Client Activation	-	-	**[7.1.3.1]**	-	-	**[7.1.3.1.2]**	**[7.1.3.1.2]**	-	-
DCD Client Deactivation	-	-	**[7.1.3.2]**	-	-	**[7.1.3.2.2]**	**[7.1.3.2.2]**	-	-
Application Registration	**[7.3.1.1]**	-	**[7.1.3.3]**	-	-	-	-	-	-
Application Deregistration	**[7.3.1.2]**	-	**[7.1.3.4]**	-	-	-	-	-	-
Channel Discovery	**[7.3.2.1]**	**[7.3.2.5]**	**[7.1.3.10.3]**	**[7.1.3.10.1]**	**[7.1.3.10.1]**	**[7.1.3.10.3]**	**[7.1.3.10.3]**	-	-
Channel Subscription	**[7.3.1.3]**	**[7.3.1.7]**	**[7.1.3.7]**	-	-	**[7.1.3.9]**	**[7.1.3.9]**	**[7.2.1.1]**	**[7.2.1.3]**
Channel Un-subscription	**[7.3.1.5]**	**[7.3.1.6]**	**[7.1.3.8]**	-	-	**[7.1.3.8.2]**	**[7.1.3.8.2]**	**[7.2.1.2]**	-
Subscription Update	**[7.3.1.4]**	-	-	-	-	-	-	-	**[7.2.1.4]**
Channel Suspend	**[7.3.2.4.1]**	**[7.3.2.4.5]**	**[7.1.3.11.1]**	-	-	**[7.1.3.11.5]**	**[7.1.3.11.5]**	**[7.2.2.1.1]**	**[7.2.2.1.5]**
Channel Resume	**[7.3.2.4.3]**	**[7.3.2.4.7]**	**[7.1.3.11.3]**	-	-	**[7.1.3.11.7]**	**[7.1.3.11.7]**	**[7.2.2.1.3]**	**[7.2.2.1.7]**
Content Delivery	**[7.3.2.2]**	**[7.3.2.2]**	**[7.1.1.1]**	**[7.1.2.1]**	**[7.1.2.1]**	**[7.1.2.2]**	**[7.1.2.2]**	**[7.2.2.2]**	**[7.2.2.3]**
Content Submission	**[7.3.2.3]**	-	**[7.1.1.2]**	-	-	-	-	**[7.2.2.2]**	-
Content Repair	-	-	**[7.1.3.12]**	-	-	-	-	-	-
Usage Tracking Report	-	-	**[7.1.3.6]**	-	-	**[7.1.3.6]**	**[7.1.3.6]**	**[7.2.1.7]**	**[7.2.1.7]**
Contextual Information Upload	-	-	**[7.1.3.5]**	-	-	**[7.1.3.5]**	**[7.1.3.5]**	-	-
Channel Registration	-	-	-	-	-	-	-	**[7.2.1.5]**	**[7.2.1.5]**
Channel Metadata Update	-	**[7.3.1.8]**	-	**[7.1.3.13]**	**[7.1.3.13]**	-	-	-	-
Channel								**[7.2.1.6]**	**[7.2.1.6]**

Deregistration									
Connection Profile Update				**[7.1.3.14]**	**[7.1.3.14]**				

Table 1 DCD Operation-Interface mapping

Editor's Notes: "-" – Not applicable

6. DCD Operations

6.1 DCD Client Operations

If the device runtime supports it, the DCD Client SHOULD NOT interfere with other device use or applications, e.g. execute cooperatively.

Upon receiving messages from the DCD Server or the DCD Enabled Client Application, the DCD Client SHALL validate the parameters of these messages as defined in Section 7 and send appropriate errors as defined in Section 13.

6.1.1 Client Activation

Upon the first DCD-Enabled Client Application to register with a particular DCD Server, a DCD Client SHALL activate the service with the DCD Server as defined by the applicable DCD-3 connection profile. If there is no DCD-3 connection profile in the Application Profile, the DCD Client SHALL activate the service with a default DCD-3 connection profile, which is set originally in the device or provisioned by the DM (e.g. OMA DM). In order to activate the service, the DCD Client SHALL send ClientActivationRequest message with Device-ID and Version, as described in Section 7.1.3.1.1. Note: the presence of a default DCD-3 connection profile in non-volatile memory ensures the ability to re-activate DCD service upon master reset/clear of the device.

If the DCD Server does not require authentication and no errors occurred, the DCD Client receives a ClientActivationResponse indicating successful activation.

If the DCD Server requires authentication, the DCD Client SHALL proceed with the applicable authentication process. If the DCD Client is successfully authenticated by the DCD Server and no errors occurred, the DCD Client receives a subsequent ClientActivationResponse indicating successful activation.

If the DCD Server responded with an ErrorNotification message with error-code “authentication error”, the DCD Client SHALL NOT automatically resend the ClientActivationRequest message.

If the DCD Server responded with an indication of a temporary failure condition showing that DCD Server is unavailable, the DCD Client SHALL re-attempt to activate the service at a later time.

If Client Activation fails because the uplink is unavailable, the DCD Client SHALL inform the DCD Enabled Client Application using the ErrorNotification message.

When the pull DCD-3 interface is unavailable because the uplink is unavailable, the DCD Client SHOULD continue to support the DCD-3 interface via Push or broadcast bearers, if available and feasible.

When the DCD-1interface is unavailable because the uplink is unavailable, the DCD Client SHOULD continue to support reception of content via the DCD-2 interfacefor subscribed channels, if feasible

In the event of successful activation, the ClientActivationResponse message contains Session-ID parameter issued by the DCD Server. The DCD Client SHALL include this Session-ID parameter in all messages sent to the DCD Server during the lifetime of the session.

If it receives a RequestForClientActivation message from the DCD Server, the DCD Client SHALL initiate service activation with a ClientActivationRequest message, as described in Section 7.1.3.1.2. If the RequestForClientActivation contains a dcd-3-connection-profile with a new dcd-server-address, i.e. not present in the DCD Client’s existing connection profiles, the DCD Client SHALL initiate service activation with the DCD Server only if the DCD Server is trusted. Note that how trust is determined is an implementation issue, e.g. DCD Clients may inherently trust the DCD Server due to trust placed in the DCD-2 interface bearer (e.g. OMA Push, which is a secured service enabler in typical deployments), may use a whitelist of trusted DCD Server domains, or may require user confirmation for the new activation.

If the Session-TTL attribute is non-zero, upon expiration of the session, the DCD Client SHALL send a new ClientActivationRequest as soon as possible.

The DCD Client MAY deactivate the DCD service at any time. To deactivate the service, the DCD Client SHALL send a ClientDeactivationRequest to the DCD Server, as described in Section 7.1.3.2. The DCD Client SHALL re-attempt to deactivate the service at the later time if deactivation request was unsuccessful.

If it receives a ClientDeactivationNotification message from the DCD Server, the DCD Client SHALL NOT attempt to reactivate the service with that DCD Server until it receives a RequestForClientActivation message from the DCD Server.

Upon deactivation of the DCD Service triggered by deregistration of the last DCD Enabled Client Application, the DCD Client SHOULD remove any remaining registration information for registered DCD Enabled Client Applications.

If the DCD Client receives a ConnectionProfileUpdate message including a modified DCD-3 connection profile from the DCD Server, the DCD Client:

- SHALL send the ConnectionProfileConfirmation to the DCD Server
- and, as soon as possible, e.g. upon completion of any pending content delivery, SHOULD
    - send a ClientDeactivationRequest message to deactivate the session with the current DCD Server. if there was a change in the DCD-3 server address
    - send a ClientActivationRequest message according to the DCD-3 connection profile to re-establish a new session

The DCD Client SHOULD support the storing or backup of subscription information and information about registered applications on removable devices (e.g. using smart cards, (U)SIM, removable disks) to allow portability of service between devices with compatible DCD Client versions of the same vendor.

When the DCD Client has detected that a change of subscriber has occurred for a device with previously activated DCD Service, the DCD Client SHOULD prevent access to DCD Content and Channel Metadata related to the previous subscriber.

The DCD Client SHOULD support subscriber selected privacy options for access to the DCD Content.

6.1.2 Application Registration

In order to allow a DCD Enabled Client Application to receive the content from the DCD enabler, a DCD Client SHALL register the application with the enabler at the applicable DCD Server to which DCD service has been previously activated. In certain broadcast scenarios where uplink to the DCD Server is not available, the registration is local to the DCD Client and the text below is not applicable.

When it receives an ApplicationRegistrationRequest message from the DCD Enabled Client Application (see Section 7.3.1.1) or it receives the Application Profile (see Section 8.1) through other means (e.g. from mediator or preconfigured) , the DCD Client SHALL send an ApplicationRegistrationRequest message to the DCD Server including the Application Profile and other parameters, as described in Section 7.1.3.3.

If the DCD Server responded with an indication of a temporary failure condition showing that the DCD Server is unavailable, the DCD Client SHALL re-attempt to register the application at a later time.

If no errors occurred, and the DCD Client receives the ApplicationRegistrationResponse message from the DCD Server, the DCD Client SHALL respond to the DCD Enabled Client Application with a corresponding ApplicationRegistrationResponse message.

If the ApplicationRegistrationResponse message from the DCD Server contains Channel Metadata (see Section 8.2) for channels successfully matched to the Application Profile preferences, the DCD Client SHALL extract the application-related subset of Channel Metadata and specify its location in the Channel-Discovery-Information parameter of the ApplicationRegistrationResponse message to the DCD Enabled Client Application.

Upon receiving an ApplicationDeregistrationRequest message from the DCD Enabled Client Application (see Section 7.3.1.2), the DCD Client SHALL respond with an ApplicationDeregistrationResponse message indicating successful deregistration. Consecutively, the DCD Client SHALL send an ApplicationDeregistrationNotification message to the DCD Server, as described in Section 7.1.3.4. If no errors occurred, the DCD Client receives the DCD Server response in the ApplicationDeregistrationConfirmation message.

If the DCD Client detects that the DCD Enabled Client Application was uninstalled without prior deregistration (e.g. upon receiving an error on content availability notification), the DCD Client SHALL send an ApplicationDeregistrationNotification message to the DCD Server, as described in Section 7.1.3.4

The DCD Client SHALL re-attempt to deregister the application with ApplicationDeregistrationNotification message at a later time if application deregistration confirmation was unsuccessful. In case of successful deregistration confirmation, the DCD Client SHOULD remove all registration information associated with the deregistered DCD Enabled Client Application.

If the DCD Client detects that the DCD Enabled Client Application did not terminate channel subscriptions prior to deregistration, the DCD Client SHALL unsubscribe on behalf of the application by sending DCD-3 ChannelUnsubscriptionRequest.

Upon deregistration of DCD Enabled Client Application, if its content storage was provided by the DCD Client, the DCD Client SHOULD remove all stored DCD Content for the deregistered application, except for content that is shared by other registered applications.

6.1.2.1 Receive-Only Client Case

For the ***receive-only client*** case:

DCD Enabled Client Application registers itself with DCD Client over DCD-CAR interface

- Application Profile
    - DCD Enabled Client Application provides DCD-3 connection profile with broadcast profile as defined in Section 8.1
- Channel Metadata including DCD-2 broadcast profile
    - DCD Enabled Client Application provides list of channels with their related DCD-2 broadcast profiles

DCD Client listens to the following channels:

- DCD-3 Admin broadcast channel
- DCD-2 Content broadcast channels

DCD Enabled Client Application deregisters itself from DCD Client over DCD-CAR interface or can be deregistered by the DCD Client when it detects that the application was uninstalled.

6.1.3 Channel Subscription

6.1.3.1 Internal Channel Subscription

The following refers to DCD Client handling of the ChannelSubscription initiated by the DCD Enabled Client Application.

Upon receiving an acceptable DCD-CAR SubscriptionRequest message (see Section 7.3.1.3) from a DCD Enabled Client Application, the DCD Client SHALL validate whether there are other DCD Enabled Client Applications on the device already subscribed to this channel. If there is another DCD Enabled Client Application already subscribed to this channel, the DCD Client SHALL validate that Channel-Metadata, Subscription-ID, and Selected-Purchase-Option parameters are compatible (i.e. matching values or subset) between the two subscriptions. If these parameters are incompatible, the behaviour of the DCD Client is implementation-specific (e.g. it can merge the preferences in the updated subscription with the DCD Server, return an error to the DCD Enabled Client Application, etc.).

If the DCD-CAR SubscriptionRequest message was not acceptable, e.g. one or more included delivery-personalization-metadata option could not be honored by the DCD Client, the DCD Client SHALL return an ErrorNotification to the DCD Enabled Client Application, with error details as appropriate.

If there's no need to update existing channel subscription with the DCD Server, the DCD Client SHALL send a DCD-CAR SubscriptionResponse (see Section 7.3.1.3) to the DCD Enabled Client Application that initiated the DCD-CAR ChannelSubscriptionRequest.

If there are no other DCD Enabled Client Applications on the device subscribed to this channel or if the subscription needs to be updated with consolidated preferences from multiple DCD Enabled Client Applications, the DCD Client SHALL send a DCD-3 ChannelSubscriptionRequest (see Section 7.1.3.7) to the DCD Server to which the application was registered. The DCD Client SHALL include in the DCD-3 ChannelSubscriptionRequest all applicable parameters from the DCD-CAR SubscriptionRequest.

Upon a receiving a DCD-3 ChannelSubscriptionResponse message (see Section 7.1.3.7) from the DCD Server, the DCD Client SHALL send a DCD-CAR SubscriptionResponse (see Section 7.3.1.3) to the DCD Enabled Client Application that initiated the DCD-CAR ChannelSubscriptionRequest. The DCD Client SHALL include in the DCD-CAR SubscriptionResponse all applicable parameters from the DCD-3 ChannelSubscriptionResponse.

In the case of successful subscription, the DCD Client SHALL store any new or updated Channel Metadata related to the subscription.

When the DCD client is BCAST and/or CBS enabled, the DCD Client SHALL also follow the rules and procedures of [DCD-TS-BCAST and/or DCD-TS-CBS] in addition to those listed above.

6.1.3.2 External Subscription

The following refers to DCD Client handling of the Channel Subscription notification from the DCD Server.

Upon receiving a DCD-3 ChannelSubscriptionNotification message (see Section 7.1.3.9) from the DCD Server, the DCD Client SHALL send a DCD-CAR SubscriptionValidationRequest (see Section 7.3.1.7) to the DCD Enabled Client Application identified by the application-id parameter in the DCD-3 ChannelSubscriptionNotification message. The DCD Client SHALL include in the DCD-CAR SubscriptionValidationRequest all applicable parameters from the DCD-3 ChannelSubscriptionNotification.

Upon receiving an acceptable DCD-CAR SubscriptionValidationResponse message (see Section 7.3.1.7) from the DCD Enabled Client Application, the DCD Client SHALL send a DCD-3 ChannelSubscriptionNotificationResponse (see Section 7.1.3.9) to the DCD Server. The DCD Client SHALL include in the DCD-3 ChannelSubscriptionNotificationResponse all applicable parameters from the DCD-CAR SubscriptionValidationResponse.

If the DCD-CAR SubscriptionValidationResponse message was not acceptable, e.g. an included delivery-personalization-metadata option could not be honored by the DCD Client, the DCD Client SHALL:

- return an ErrorNotification to the DCD Enabled Client Application, with error details as appropriate
- send a DCD-3 ChannelSubscriptionNotificationResponse message, with Subscription-Declined set to "True"

In the case of successful subscription validation, the DCD Client SHALL store any new or updated Channel Metadata related to the subscription.

When the DCD client is BCAST and/or CBS enabled, the DCD Client SHALL also follow the rules and procedures of [DCD-TS-BCAST and/or DCD-TS-CBS] in addition to those listed above.

6.1.3.3 Channel Subscription Update

The following refers to DCD Client handling of the Channel Subscription Update initiated by the DCD Enabled Client Application.

Upon receiving an acceptable DCD-CAR SubscriptionUpdateRequest message (see Section 7.3.1.4) from a DCD Enabled Client Application, the DCD Client SHALL:

- update stored Channel Metadata parameters (see Section 8.2) relevant to the DCD Client with the parameters provided in the DCD-CAR SubscriptionUpdateRequest

- validate whether there are other DCD Enabled Client Applications on the device already subscribed to this channel. If there is another DCD Enabled Client Application already subscribed to this channel, the DCD Client SHALL validate that Channel-Metadata parameters are compatible (i.e. matching values or subset) between the two subscriptions. If these parameters are incompatible, the behaviour of the DCD Client is implementation-specific (e.g. it can merge the preferences in the updated subscription with the DCD Server, return an error to the DCD Enabled Client Application, etc.).
- send DCD-3 ChannelSubscriptionRequest to the DCD Server if DCD-CAR SubscriptionUpdateRequest message contained updated Channel Metadata parameters relevant to the DCD Server or if relevant Channel Metadata parameters need to be updated at the DCD Server as a result of subscription consolidation from two or more DCD Enabled Client Applications. The DCD Client SHALL include in the DCD-CAR SubscriptionResponse all applicable parameters from the DCD-3 ChannelSubscriptionResponse. The DCD Client SHOULD remove from the Channel Metadata sent to the DCD Server all parameters that are relevant only to the DCD Client.
- send DCD-CAR SubscriptionUpdateResponse (see Section 7.3.1.4) to the DCD Enabled Client Application if DCD-CAR SubscriptionUpdateRequest message didn't contain updated Channel Metadata parameters relevant to the DCD Server

If the DCD-CAR SubscriptionUpdateRequest message was not acceptable, e.g. an included delivery-personalization-metadata option could not be honored by the DCD Client, the DCD Client SHALL return an ErrorNotification to the DCD Enabled Client Application, with error details as appropriate.

Upon receiving a DCD-3 ChannelSubscriptionResponse message (see Section 7.1.3.7) from the DCD Server, the DCD Client SHALL:

- update stored Channel Metadata parameters (see Section 8.2) relevant to the DCD Client with the parameters provided in the DCD-3 ChannelSubscriptionResponse message, using the comparison procedure described for Channel Metadata Update (see Section 6.1.4)
- send a DCD-CAR SubscriptionUpdateResponse (see Section 7.3.1.4) to the DCD Enabled Client Application that initiated the DCD-CAR SubscriptionUpdateRequest. The DCD Client SHALL include in the DCD-CAR SubscriptionUpdateResponse all applicable parameters from the DCD-3 ChannelSubscriptionResponse.

When the DCD client is BCAST and/or CBS enabled, the DCD Client SHALL also follow the rules and procedures of [DCD-TS-BCAST and/or DCD-TS-CBS] in addition to those listed above.

6.1.3.4 Channel Unsubscription

The following refers to DCD Client handling of the Channel Unsubscription transactions.

Upon receiving a DCD-CAR UnsubscriptionRequest message (see Section 7.3.1.5) from a DCD Enabled Client Application, the DCD Client SHALL send a DCD-3 ChannelUnsubscriptionRequest to the DCD Server to which the application was registered if there are no other DCD Enabled Client Applications on device subscribed to this channel. The DCD Client SHALL include in the DCD-3 ChannelUnsubscriptionRequest all applicable parameters from the DCD-CAR UnsubscriptionRequest. If there are other DCD Enabled Client Applications on device subscribed to this channel, the DCD Client SHALL send a DCD-CAR UnsubscriptionResponse to the DCD Enabled Client Application that issued the DCD-CAR ChannelUnsubscriptionRequest.

Additionally, the DCD Client SHALL send DCD-3 ChannelUnsubscriptionRequest to the DCD Server upon discovery that the last DCD Enabled Client Application subscribed to a channel was unregistered or uninstalled without prior unsubscription.

Upon a receiving a DCD-3 ChannelUnsubscriptionResponse message (see Section 7.1.3.8) from the DCD Server the DCD Client SHALL send a DCD-CAR UnsubscriptionResponse to the DCD Enabled Client Application that issued the DCD-CAR UnsubscriptionRequest if the transaction was initiated by a DCD Enabled Client Application.

In the case of successful unsubscription, the DCD Client SHALL remove any Channel Metadata related to the terminated subscription, except for Channel Guide information.

Upon receiving a DCD-3 ChannelUnsubscriptionNotification message (see Section 7.1.3.8.2) from the DCD Server, the DCD Client SHALL send a DCD-CAR UnsubscriptionNotification (see Section 7.3.1.6) to all DCD Enabled Client Applications subscribed to this channel. The DCD Client SHALL remove any Channel Metadata related to the terminated subscription.

When the DCD client is BCAST and/or CBS enabled the DCD Client SHALL also follow the rules and procedures of [DCD-TS-BCAST and/or DCD-TS-CBS] in addition to those listed above.

Upon receiving a DCD-3 ChannelUnsubscriptionNotification message (see Section 7.1.3.8.2) via a point to point bearer, the DCD Client SHALL respond with a ChannelUnsubscriptionConfirmation.

6.1.3.5 Receive-Only Client Case

In the **Receive-Only Client** case, the DCD Client locally subscribes or unsubscribes as a result of a DCD Enabled Client Application subscribe or unsubscribe request over the DCD-CAR interface. In such cases the DCD Client starts or stops listening to the specific channel, and responds with the appropriate message to the DCD Enabled Client Application.

6.1.4 Channel Metadata Update

The following refers to DCD Client handling of the DCD-3 ChannelMetadataUpdate message, as sent by the DCD Server to inform about the changes to previously received Channel Metadata.

Upon receiving a ChanneMetadataUpdate message, as described in Section 7.1.3.13, from a DCD Server, the DCD Client SHALL compare the received ChannelMetadataUpdate message with the existing Channel Metadata for the same Channel, and

- If a Channel Metadata field in the existing Channel Metadata is not present in the ChannelMetadataUpdate message, the DCD Client SHALL use the existing value for this field
- If a Channel Metadata field is empty in the ChannelMetadataUpdate message, the DCD Client SHALL remove the previous value for this field from the existing Channel Metadata
- If a Channel Metadata field is present in the ChannelMetadataUpdate message, the DCD Client SHALL replace the data in the same field in the existing Channel Metadata

If the update affects the Channel Metadata fields relevant to a DCD Enabled Client Application, the DCD Client SHALL inform the DCD Enabled Client Application by sending a ChannelMetadataUpdate message according to Section 7.3.1.8.

Upon successful processing of Channel Metadata, following reception of a ChannelMetadataUpdate message as described in Section 7.1.3.13, the DCD Client SHALL send a ChannelMetadatalUpdateConfirmation message according to Section 7.1.3.13 to the DCD Server.

When the DCD client is BCAST and/or CBS enabled, the DCD Client SHALL also follow the rules and procedures of [DCD-TS-BCAST and/or DCD-TS-CBS] in addition to those listed above.

6.1.5 Content Delivery

6.1.5.1 Receive-Only Client Case

For the ***receive-only client*** case:

DCD Client listens to channel based on parameters specified in DCD-2 connection profile

DCD Client receives content and delivers the received content to the DCD Enabled Client Application via DCD-CADE.

6.1.5.2 Automatic Content Update

The following refers to DCD Client handling of the DCD-1 ContentUpdateRequest message, as sent for automatic update of specific channels. Automatic update can occur upon a fixed schedule or when all channel content has expired, as described below.

DCD Clients SHALL automatically request content updates under the following conditions:

- If the pull-schedule channel metadata attribute is specified, DCD Clients SHALL automatically request a content update per the indicated schedule.
- If the update-upon-expiration channel metadata attribute is “True”, DCD Clients SHALL automatically request a content update when all content for the channel has expired.

DCD Clients SHALL NOT automatically request content updates as described above if prohibited from doing so, e.g.:

- If the deliver-when-roaming channel metadata attribute is “False”, DCD Clients SHALL NOT automatically request content updates if they are aware that the user is being served by a domestic roaming network, or is internationally roaming.
- If the DCD Client has suspended delivery for a channel, the DCD Client SHALL NOT automatically request content updates for that channel.

DCD Clients SHOULD send a single ContentUpdateRequest message for update of multiple channels if the channels are due for update at the same time and the same DCD-1 connection profile settings are applicable for these channels

DCD Clients SHALL send automatic ContentUpdateRequest messages to the DCD Server per the applicable dcd-1-connection-profile.

DCD Clients SHOULD delay automatic content update requests by a pseudo-random value based upon a unique device attribute (i.e. different from other devices).

If the network-preferences channel metadata attribute is specified, the DCD Client SHALL send the ContentUpdateRequest message as specified. If there are multiple networks specified in the network-preferences and multiple networks are available for use by the DCD Client, the DCD Client SHALL send the ContentUpdateRequest message via the highest priority network as indicated by the network-preferences attribute, if applicable.

If the ContentUpdateResponse message from the DCD Server was accepted, further handling is described in Section 6.1.5.6.

6.1.5.3 On-Demand Content Update

The following refers to DCD Client handling of the DCD-1 ContentUpdateRequest message, as sent for on-demand update of specific channels or retrieval of specific content items.

If the on-demand-pull-allowed channel metadata attribute is “True” and use of the DCD-1 interface is allowed for the channel (i.e. the dcd-interface channel metadata attribute contains a “DCD-1” value), the DCD Client SHALL accept the DCD-CADE ContentRequest message as an on-demand content update request from DCD Enabled Client Applications.

If the on-demand-pull-allowed channel metadata attribute is “False” or use of the DCD-1 interface is not allowed for the channel, the DCD Client SHALL respond with an ErrorNotification message with error-code “not allowed” to on-demand content update requests from DCD Enabled Client Applications.

The DCD Client SHALL accept requests for specific content items by content-id; if a particular requested content item is already available in the DCD-managed storage on the device, the DCD Client SHALL return the content to the DCD Enabled Client Application.

If a channel is suspended, the DCD Client SHALL respond with an ErrorNotification message with error-code “temporary failure” to on-demand content update requests from DCD Enabled Client Applications.

In creating ContentUpdateRequest messages for on-demand content updates, DCD Clients SHALL include:

- the session-id for the active DCD session with the DCD Server via which the channel was subscribed
- a new message-id value
- one of the following, if specified by the DCD Enabled Client Application in the ContentRequest message
    - a requested channel-id
    - a requested content-id
    - a requested content-address

DCD Clients SHALL send on-demand ContentUpdateRequest messages to the DCD Server per the applicable dcd-1-connection-profile.

If the network-preferences channel metadata attribute is specified, the DCD Client SHALL send the ContentUpdateRequest message as specified. If there are multiple networks specified in the network-preferences and multiple networks are available for use by the DCD Client, the DCD Client SHALL send the ContentUpdateRequest message via the highest priority network as indicated by the network-preferences attribute, if applicable.

If the ContentUpdateResponse message from the DCD Server was accepted, further handling is described in Section 6.1.5.6.

6.1.5.4 Content Update Push

DCD Clients SHALL discard any content unrelated to channels subscribed by registered DCD Enabled Client Applications on the device, and:

- SHALL send an ErrorNotification message with error-code “channel not subscribed” to the DCD Server..

If the DCD-2 ContentUpdatePush message was accepted, further handling is described in Section 6.1.5.6.

In the case that Cell Broadcast or BCAST are used, further handling is described in [DCD-TS-BCAST] or [DCD-TS-CBS] as applicable.

6.1.5.5 Content Update Notification

If the dcd-content-reference message element is included in the message, the DCD Client SHALL send a DCD-3 ContentUpdateRequest message to the DCD Server at the address indicated by the dcd-content-reference message attribute.

If content-metadata message element is included, the DCD Client SHALL proceed with handling of the content items as described in Section 6.1.5.6.

DCD Clients SHALL discard any content unrelated to channels subscribed by registered DCD Enabled Client Applications on the device, and:

- SHALL send an ErrorNotification message with error-code “channel not subscribed” to the DCD Server.

If the ContentUpdateNotification message from the DCD Server was accepted, further handling is described in Section 6.1.5.6.

In the case that Cell Broadcast or BCAST are used, further handling is described in [DCD-TS-BCAST] or [DCD-TS-CBS] as applicable.

6.1.5.6 Content Item Handling

The following refers to DCD Client handling of content items received via the DCD-1 ContentUpdateResponse message, the DCD-2 ContentUpdatePush message, or the DCD-2 ContentUpdateNotification message.

DCD Clients SHALL store the content metadata, if updated, for later use or delivery to DCD Enabled Client Applications.

For channels with DCD-managed storage, DCD Clients SHALL store content items, if updated, and associated content metadata for later use or delivery to DCD Enabled Client Applications.

If there is inadequate DCD-managed storage to store the received content for a channel, the DCD Client SHALL

- send a ContextualInformationUpload message to the DCD Server, specifying the free-storage attribute, according to the defined policy per [Section 7.1.3.5]

For channels for which the content-storage-location channel metadata attribute is set, DCD Clients SHALL store content items and the application-relevant subset of associated content metadata, if updated, at the indicated location [see Section 8.3].

If the replaces-content-id content metadata attribute is set for a content item, the DCD Client SHALL replace the indicated content item and associated metadata with the new content item and associated metadata at the applicable storage location.

If the delivery-priority content metadata attribute is set to "high" for a content item, the DCD Client SHOULD expedite storage and content availability notification actions.

Note that beyond expedited handling, actions upon content items with the content metadata attribute "emergency-content" set to "True" are implementation-specific.

If the content-price content metadata element is not present for a content item:

- If the actual content item was not included in the message, but is referenced by the content-address content metadata attribute only, the DCD Client SHALL send a DCD-1 ContentUpdateRequest message to the DCD Server at the indicated content-address.
- If the actual content item was not included in the message, but is referenced by the content-id content metadata attribute only, the DCD Client SHALL send a DCD-1 ContentUpdateRequest message to the DCD Server, including content-id.
- If the aux-content-link content metadata attribute is present, the DCD Client SHALL send a DCD-1 ContentUpdateRequest message to the DCD Server at the indicated aux-content-link.

The DCD Client MAY choose not to retrieve the content automatically (as described above) if undesired based on content metadata received in the message (e.g. due to content-length, parental-rating, etc.)

If the DCD Client chooses not to retrieve a content item, it SHALL NOT send DCD-1 ContentUpdateRequest to the DCD Server.

DCD Clients SHALL send ContentUpdateRequest messages for content item or aux-content retrieval to the DCD Server per the applicable dcd-1-connection-profile.

If the network-preferences attribute is specified in channel or content metadata, the DCD Client SHALL send the ContentUpdateRequest message as specified. If there are multiple networks specified in the network-preferences and multiple networks are available for use by the DCD Client, the DCD Client SHALL send the ContentUpdateRequest message via the highest priority network as indicated by the network-preferences attribute, if applicable.

If the content-encoding attribute specified in content metadata, the DCD Client SHALL decode (e.g. extract) the content item accordingly prior to delivery to the DCD Enabled Client Application.

If the dcd-provided-storage-size channel metadata attribute is set, the DCD Client SHALL store received content in its reserved storage. Otherwise, if the content-storage-location channel metadata attribute is set, the DCD Client SHALL store received content in the indicated location. Note: use of the content-storage-location attribute also enables manual deletion of outdated content through the user interface of the DCD-Enabled Client Application which manages the content.

If the content-expiration attribute is specified in content metadata and the content item is expired according to this attribute, the DCD Client SHALL disregard this content item. If this content item has already been delivered to the device and stored in the DCD managed storage, the DCD Client SHALL remove it from the storage.

If the content-expiration content metadata attribute is not specified for a content item stored in DCD-managed storage, the DCD Client SHALL use storage management rules from the content-expiration channel metadata attribute, if specified.

To reliably support the content-expiration attribute, the DCD Client SHOULD support UTC time synchronization with network time sources.

If the content-availability-notification channel metadata attribute is "True" or the content-price attribute is specified, DCD Clients SHALL send the DCD-CADE Content message to applicable DCD Enabled Client Applications, including:

- for each content item that has been received (as part of the initial message or after retrieval at the content-address), one of
    - the content-storage-location content metadata attribute for each content item, set to the location of the content in DCD-managed storage or DCD Enabled Client Application specified storage (i.e. per the content-storage-location channel metadata attribute provided during channel subscription)
    - the actual content item, if neither dcd-provided-storage-size or content-storage-location are set
- the application-related subset of content metadata attributes as received for each content item, e.g.: content-id, channel-id, mime-type, content-length, and content-name [see Section 8.3 for details].

If one or more content items have a content-delivery-notification content metadata attribute set to "True", the DCD Client SHALL send the DCD-1 ContentDeliveryConfirmation message to the DCD Server after content item handling is complete, including:

- the session-id from the ContentUpdateResponse or ContentUpdatePush
- a new message-id value
- the content-confirmation attribute including the content-id and delivery status of each content item for which content delivery notification is required, and for which handling has been completed since the last ContentDeliveryConfirmation message

The state of target DCD Enabled Client Application SHOULD NOT affect the DCD Client's ability to deliver notifications (e.g. subject to device runtime limitations, the DCD Client can launch an inactive application, "wake up" a dormant application, or pass notification to an active application). If the DCD Client can only notify active applications it SHOULD deliver content availability notification as soon as it detects that the target application became active.

DCD Client SHALL follow the packaging rules as defined in Section 9.1.

In the broadcast case, the DCD Client MAY discard any content and content metadata inconsistent with the Content Types channel metadata attribute for subscribed DCD Enabled Client Applications.

6.1.6 Content Submission

Upon reception of the DCD-CADE ContentSubmitRequest message (see Section 7.3.2.3), if use of the DCD-1 interface is allowed for this channel (i.e. the dcd-interface channel metadata attribute contains a "DCD-1" value), DCD Clients SHALL send the DCD-1 ContentSubmitRequest message to the DCD Server. The DCD Client SHALL include in the DCD-1 ContentSubmitRequest the Submit-Package parameter from the DCD-CADE ContentSubmitRequest message and other parameters as described in Section 7.1.1.2).

DCD Clients SHALL send ContentSubmitRequest messages to the DCD Server per the applicable dcd-1-connection-profile.

If the network-preferences attribute is specified in channel or content metadata, the DCD Client SHALL send the ContentSubmitRequest message as specified. If there are multiple networks specified in the network-preferences and multiple networks are available for use by the DCD Client, the DCD Client SHALL send the ContentSubmitRequest message via the highest priority network as indicated by the network-preferences attribute, if applicable.

Upon reception of the DCD-1 ContentUpdateResponse message, the DCD Client SHALL:

- send the DCD-1 ContentDeliveryConfirmation message (see Section 7.1.1.1) to the DCD Server if one or more content items in the Content-Package have a content-delivery-notification Content Metadata attribute set to "True"
- send the DCD-CADE Content message (see Section 7.3.2.2) to the DCD Enabled Client Application.

Upon reception of the DCD-1 ContentSubmitConfirmation message, the DCD Client SHALL:

- send the DCD-CADE ContentSubmitConfirmation (see Section 7.3.2.3) to the DCD Enabled Client Application

6.1.7 Channel Suspension

6.1.7.1 Suspension Effect Upon DCD Transactions

When a specific channel is suspended for all subscribed DCD-Enabled Client Applications, the DCD Client SHALL NOT:

- issue ContentUpdateRequests for this channel
- respond to ContentUpdateNotification for this channel, with ContentUpdateRequests

The previous requirement applies when:

- a suspended channel is subscribed by only one DCD-Enabled Client Application, or
- all subscribed DCD-Enabled Client Applications have suspended the channel,
- the DCD Client has suspended delivery for all DCD-Enabled Client Applications, e.g. due to the channel metadata deliver-when-roaming attribute.

When a channel is suspended for a specific DCD-Enabled Client Application, the DCD Client SHALL NOT deliver Content notifications to that DCD-Enabled Client Application.

When all suspendable channels (i.e. suspend-allowed = “True”)are suspended for all DCD Enabled Client Applications, the DCD Client SHALL NOT:

- issue ChannelDiscoveryRequests
- automatically issue UsageTrackingReport

When a previously suspended DCD Channel is resumed, the DCD Client SHALL deliver any buffered content that has not yet been delivered to applicable DCD Enabled Client Applications.

6.1.7.2 Channel Suspension Requested by DCD Client

A DCD Enabled Client Application can initiate channel suspension by sending the DCD-CADE ChannelSuspendRequest message with the Channel IDs of the channels to be suspended.

If the suspend-allowed channel metadata attribute is set to “False” for the channel, DCD Clients SHALL reject DCD-CADE ChannelSuspendRequest messages, by sending the DCD-CADE ErrorNotification message with error code “not allowed” [Section 13].

If the channel to be suspended is shared with other DCD Enabled Client Applications that have not requested suspension, the DCD Client SHALL process suspension locally, such as:

- SHALL NOT send a suspension request to the DCD Server
- SHALL send a DCD-CADE ChannelSuspendResponse message to the DCD Enabled Client Application.

Upon channel suspension by the DCD Client, if the DCD Client is in the process of a content delivery transaction, it SHOULD terminate the transaction locally (e.g. by requesting reset of the HTTP connection), if technically feasible, otherwise the DCD Client SHALL continue with the transaction until the transaction completes or is terminated by the DCD Server.

If the transaction is terminated before the content has been fully delivered, i.e. upon local termination of the transaction or early termination of the transaction by the DCD Server, the DCD Client SHALL report the error as described in [Section 13].

When the suspension conditions in the Channel Metadata are satisfied (e.g. when device is roaming and deliver-when-roaming is set to “False”), the DCD Client can initiate channel suspension. If these suspension conditions are in the delivery personalization metadata submitted by a DCD Enabled Client Application and there are other DCD Enabled Client

Applications subscribed to this channel, the DCD Client SHALL process suspension locally (as described above), unless these conditions are satisfied for all applications subscribed to this channel.

The DCD Client SHALL initiate channel suspension by sending the DCD-3 ChannelSuspendRequest message with the target Channel IDs.

When the DCD Client receives a ChannelSuspendResponse from the DCD Server, it SHALL send the DCD-CADE ChannelSuspendResponse to any DCD Enabled Client Application waiting for the confirmation.

After DCD Enabled Client Application-initiated suspension is completed, only the DCD Enabled Client Application can resume content delivery. When determined necessary (e.g. upon request by the user or application-specific event), the DCD Enabled Client Application can resume content delivery by sending the DCD-CADE ChannelResumeRequest message. When the DCD Client receives the ChannelResumeRequest message, it SHALL send the DCD-3 ChannelResumeRequest message to the DCD Server with the target Channel IDs, unless the suspension was processed locally (as described above).

Resumption of content delivery by one DCD Enabled Client Application SHALL NOT result in resumption for all DCD Enabled Client Applications, i.e. each shall request resumption individually, if it had requested suspension individually.

When the resumption conditions in the Channel Metadata are satisfied (e.g. in the event of return to home network when deliver-when-roaming is set to “False”), and there are DCD Enabled Client Applications for which the channel should be resumed based on these resumption conditions, the DCD Client SHALL initiate channel resumption by sending the DCD-3 ChannelResumeRequest message with the target Channel IDs, unless the suspension was processed locally (as described above).

When the DCD Client receives a ChannelResumeResponse from the DCD Server, it SHALL:

- send the ChannelResumeResponse message to each applicable (as defined above) DCD Enabled Client Application, and
- resume normal DCD transaction handling for the related channels.

If some content delivery failed due to channel suspension, the DCD Client MAY send the ContentRepairRequest.

6.1.7.3 Channel Suspension Notification by DCD Server

Upon the ChannelSuspendNotification message from the DCD Server, the DCD Client SHALL

- notify any DCD Enabled Client Applications for which the channel is not already suspended, via the DCD-CADE ChannelSuspendNotification message, and
- suspend normal DCD transaction handling for the related channels, e.g. per the channel schedule or in response to ContentUpdateNotification.

Upon channel suspension by the DCD Server, if the DCD Client is in the process of a content delivery transaction, the DCD Client SHALL continue with the transaction until the transaction completes or is terminated by the DCD Server.

Upon the ChannelResumeNotification message from the DCD Server, the DCD Client SHALL:

- notify any DCD Enabled Client Applications for which the channel should be resumed, via the DCD-CADE ChannelResumeNotification message, and
- resume normal DCD operations for the related channels.

6.1.8 Channel Discovery

6.1.8.1 Channel Discovery Request to the DCD Server

DCD Clients SHALL send a ChannelDiscoveryRequest message to the DCD Server if:

- DCD-CADE ChannelDiscoveryRequest message is received from the DCD Enabled Client Applications
- ChannelDiscoveryNotification has been received from the DCD Server.

DCD Clients SHALL send ChannelDiscoveryRequest message to the DCD Server per the applicable dcd-3-connection-profile.

6.1.8.2 Channel Discovery Info from the DCD Server

Upon the DCD-3 ChannelDiscoveryInfo message, if Channels-removed element contains "*" and Channels-added element contains a list of channel metadata for added channels, the DCD Client SHALL handle it as a replacement for Channel Guide and replace the previous Channel Guide with the list of channels in Channels-added element.

Channel metadata specified in Channels-removed or Channels-added elements SHALL contain at least the channel-id attribute.

Channel metadata specified in Channels-updated element SHALL contain at least the channel-id and one other metadata attribute.

When the DCD-3 ChannelDiscoveryInfo message is received by the DCD Client and if the channel-discovery-notification attribute in the Application Profile is "True" (see Section 8.1), the DCD Client SHALL send DCD-CADE ChannelDiscoveryInfo message with appropriate (for DCD Enabled Client Application, as specified in Section 8.2.2.2) subsets of metadata from Channels-removed, Channels-added, and Channels-updated elements of the DCD-3 ChannelDiscoveryInfo message. The DCD Client SHALL send the DCD-CADE ChannelDiscoveryInfo message to all DCD-Enabled Client Applications indicated by the matching-applications attribute of the DCD-3 ChannelDiscoveryInfo message. Further actions by the DCD Client (beyond notification of DCD-Enabled Client Applications per the matching-applications attribute) are implementation or deployment specific.

When DCD-3 ChannelDiscoveryInfo message is received in the DCD Client and if the channel-discovery-notification attribute in the Application Profile is "False" or the runtime environment is not capable of sending a message to the DCD Enabled Client Application, the DCD Client SHOULD store the changes to the Channel Guide until DCD-CADE ChannelDiscoveryRequest message is received by the DCD Client.

When DCD-3 ChannelDiscoveryInfo message is received in the DCD Client and the message was triggered by DCD-CADE ChannelDiscoveryRequest message, the DCD Client SHALL send DCD CADE ChannelDiscoveryResponse message with the channel metadata subsets from Channels-removed, Channels-added, and Channels-updated elements relevant to the DCD Enabled Client Application.

If channel metadata attribute subscription-required is "True", the DCD Client SHALL auto subscribe to this channel. If the channel discovery action was delivered via broadcast, the subscription is local to the DCD Client and the DCD Server does not expect a subscription action over DCD-3.

6.1.8.3 Channel Discovery Response to the DCD Enabled Client Application

ChannelDiscoveryResponse message SHALL be sent to the DCD Enabled Client Application:

- Immediately when the DCD Client has received DCD-CADE ChannelDiscoveryRequest message and a ChanneDiscoveryInfo data is waiting for this application-id in the DCD Client.
- After the DCD Client successfully completes the DCD-3 ChannelDiscoveryRequest and DCD-3 ChannelDiscoveryInfo transactions with the DCD Server.

6.1.8.4 Channel Discovery Notification from the DCD Server

Upon reception of the DCD-3 ChannelDiscoveryNotification message, DCD Client SHALL send a ChannelDiscoveryRequest message to the DCD Server.

6.1.9 Usage Tracking Report

The DCD Client sends a UsageTrackingReport message to the DCD Server to report the DCD Client's usage tracking statistical information. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive only terminal).

The UsageTrackingReport message SHALL be sent from the DCD Client to the DCD Server as a result of one of the following two conditions:

- The DCD Client received a RequestForUsageTrackingReport message from the DCD Server
- Time to report the usage tracking statistics (if any) according to predefined schedule as per DCD Service Provider policy

If the usage-report-server-address attribute is present in the RequestForUsageTrackingReport, then the DCD Client SHALL send the UsageTrackingReport message to the specified address.

The DCD Client SHALL reset its internal counters for usage tracking after reporting current usage, i.e. in response to an on-demand usage tracking report, or after sending a policy-based usage tracking report.

The DCD Client SHALL reset its internal counters for usage tracking when establishing a new usage tracking report policy. The DCD Client SHALL NOT reset its internal counters for usage tracking when updating an existing usage tracking report policy, e.g. for modification of the reporting schedule.

The usage information elements that are monitored and reported in the device may depend upon the agreement between the device/client vendor and service provider. Example information to be considered for usage tracking attributes is as follows:

- Consumption of specific content items, per the DCD Service Provider policy

The UsageTrackingReport message SHALL be created with the attributes as defined in Section 7.1.3.6.1.

If all DCD Channels are suspended, the DCD Client SHALL not send UsageTrackingReport messages until at least one DCD Channel is resumed. Upon resumption of at least one DCD Channel, the DCD Client SHALL send the current usage tracking statistical information to the DCD Server.

If the DCD Client cannot respond to the DCD Server with the UsageTrackingReport message when one of the above mentioned respond conditions are met, the DCD Client SHALL send an error message with error-code “not allowed” as defined in Section 13.

6.1.10 Contextual Information Upload

The DCD Client sends a ContextualInformationUpload message to the DCD Server to report the DCD Client’s contextual information. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive only terminal).

The ContextualInformationUpload message SHALL be sent from the DCD Client to the DCD Server as a result of one of the following two conditions:

- The DCD Client received a ContextualInformationUploadRequest message from the DCD Server
- An automatic report is required per a predefined DCD Service Provider policy

The contextual information that is monitored and reported in the device may depend upon the agreement between the device/client vendor and service provider. Example characteristics to be considered for contextual information attributes are as follows:

- Device capabilities (e.g. available storage, current language, supported languages etc.)
- Device status (e.g. roaming status and PLMN code)

The ContextualInformationUpload message SHALL be created with the attributes as defined in Section 7.1.3.5.1

Upon finalization of Client Activation the DCD Client SHOULD send a ContextualInformationUpload message to the DCD Server to report upon the initial DCD Client contextual information.

If all DCD Channels are suspended, the DCD Client SHALL not send ContextualInformationUpload messages until at least one DCD Channel is resumed. Upon resumption of at least one DCD Channel the DCD Client SHALL send the current contextual information to the DCD Server.

If the DCD Client cannot respond to the DCD Server with the ContextualInformationUpload message when one of the above mentioned respond conditions are met, the DCD Client SHALL send an error message with error-code “not allowed” as defined in Section 13.

6.1.11 Charging

The following refers to DCD Client handling of charging as described in [DCD-TS-Charging]

When the DCD Client receives purchase options in the DCD-3 ApplicationRegistrationResponse message, the DCD-3 ChannelDiscoveryInfo message or the DCD-3 ChannelSubscriptionNotification message, it SHALL forward them to the appropriate DCD-Enabled Client Application

If purchase option metadata were specified for a channel, the DCD Client SHALL respond with an ErrorNotification (error-code “Invalid Parameter”, errored-parameter “selected-purchase-option-id”) to any DCD-CAR SubscriptionRequest message that does not contain a Selected-Purchase-Option-Id parameter.

If the DCD-CAR SubscriptionValidationRequest contained purchase option metadata, the DCD Client SHALL respond with an ErrorNotification (error-code “invalid parameter” and errored-parameter “purchase option”) to any DCD-CAR SubscriptionValidationResponse that does not contain a Selected-Purchase-Option-Id parameter.

When the DCD Client receives a Selected-Purchase-Option-Id parameter from a DCD-Enabled Client Application, it SHALL forward it to the DCD Server.

6.1.12 Content Repair

The DCD Client SHALL send DCD-3 ContentRepairRequest message as described in Section 7.1.3.12 when a content item is only partially received as a result of interrupted content delivery. The DCD Client can initiate ContentRepairRequest for requesting repair of a content item or an entire DCD-1 ContentUpdateResponse or DCD-2 ContentUpdatePush message.

If the DCD Client only received a partial part of a DCD-1 ContentUpdateResponse or DCD-2 ContentUpdatePush message and cannot identify the Content-ID of the partially received content item, then it SHALL initiate ContentRepairRequest including the Failed-Message-ID set to the Message-ID of the failed message. When requesting repair for a message, the Failed-Content-ID and Channel-ID SHALL NOT be present.

If the DCD Client only received a partial part of a content item in a DCD-1 ContentUpdateResponse or DCD-2 ContentUpdatePush message and was able to parse the content metadata, then it SHALL initiate ContentRepairRequest with the Failed-Content-ID set to the Content-ID of the failed content item and the Channel-ID that the content item belonged to. When requesting repair for a content item the Failed-Message-ID SHALL NOT be set.

For both repair instances, the total received bytes for either the message or the content item SHALL be present respectively.

6.2 DCD Server Operations

Upon receiving messages from the DCD Client or the Content Provider, the DCD Server SHALL validate the parameters of these messages as defined in Section 7 and send appropriate errors as defined in Section 13.

6.2.1 Client Activation

In order to facilitate delivery of the DCD Content to a mobile device, the DCD Server SHALL activate the service for the DCD Client on this device. To facilitate the activation, the Service Provider SHOULD provision a default DCD-3 connection profile to the DCD Client using DM (e.g. OMA DM), or ensure that devices are pre-configured with a default DCD-3 connection profile.

Upon receiving the ClientActivationRequest message from the DCD Client, the DCD Server SHALL respond to the DCD Client with the ClientActivationResponse message. If no authentication is required, and the activation request is valid, the DCD Server SHALL send to the DCD Client the ClientActivationResponse message indicating successful activation, providing the Session-ID parameter and other parameters, as described in Section 7.1.3.1.1.

If service establishment requires authentication, the DCD Server SHALL proceed with the applicable authentication process. If the DCD Client is successfully authenticated by the DCD Server, the DCD Server SHALL respond to the DCD Client sending the ClientActivationResponse message indicating successful activation, and providing the Session-ID parameter and other parameters, as described in Section 7.1.3.1.1.

The DCD Server MAY trigger DCD Client activation by sending the RequestForClientActivation message, as described in Section 7.1.3.1.2.

The DCD Server MAY deactivate the DCD Client at any time. In order to deactivate the DCD Client, the DCD Server SHALL send the ClientDeactivationNotification message to the DCD Client, as described in Section 7.1.3.2.2.

The DCD Server SHALL allow the DCD Client to deactivate the service. Upon receiving from the DCD Client the ClientDeactivationRequest message (see Section 7.1.3.2), the DCD Server SHALL respond to the DCD Client with the ClientDeactivationResponse message. If the service can be deactivated immediately the DCD Server SHALL return ClientDeactivationResponse message. Otherwise, the DCD Server SHALL return error.

If the DCD Service was deactivated by the DCD Client, the DCD Server SHALL NOT attempt to reactivate the service with the RequestForClientActivation message.

Upon deactivation of the DCD Service, the DCD Server SHALL invalidate the session and remove all registration and subscription information for the DCD Enabled Client Applications associated with the deactivated DCD Client.

If the Service Provider needs to modify the DCD-3 connection profile, the DCD Server SHALL send the ConnectionProfileUpdate message to the DCD Client.

6.2.2 Application Registration

When the DCD Server receives the ApplicationRegistrationRequest message from the DCD Client (see Section 7.1.3.3), it SHALL process the Application-Profile parameter (see Section 8.1), and match it with the Channel Metadata (see Section 8.2.2) of registered DCD Channels.

The DCD Server SHALL respond to the DCD Client with the ApplicationRegistrationResponse message.

If the Application Profile contained dcd-channel-selection-metadata information, the DCD Server SHALL provide Channel-Metadata for registered channels successfully matched to the Application Profile preferences.

If the DCD Server received the ApplicationRegistrationResponse message with the value of the Application-ID in the Application-Profile parameter matching an already registered application, the DCD Server SHALL process this message as an Application Profile update and respond with the ApplicationRegistrationResponse message, as specified above.

Upon receiving the ApplicationDeregistrationNotification message from the DCD Client (see Section 7.1.3.4), the DCD Server SHALL respond with the ApplicationDeregistrationConfirmation message. If the application can be deregistered immediately the DCD Server SHALL return the ApplicationDeregistrationConfirmation message. Otherwise, the DCD Server SHALL return error.

Any message from the DCD Server that contains the application-ID for a DCD Enabled Client Application not registered with the DCD enabler SHALL result in the DCD Client sending an error message to the DCD Server. In particular, if a message from the DCD Server related to the particular DCD Enabled Client Application arrives to the DCD Client after it sent the ApplicationDeregistrationNotification for this application, the DCD Client SHALL return an error message to the DCD Server.

Upon application deregistration, the DCD Server MAY issue the UnsubscriptionNotification messages (see Section 7.2.1.2) to Content Providers for the outstanding subscriptions, if required.

6.2.3 Channel Registration

When received the DCD-CPR ChannelRegistrationRequest message (see Section 7.2.1.5), the DCD Server SHALL

- process and validate Channel-Metadata specified in the message
- issue Channel-ID for this channel

The DCD Server SHOULD respond with DCD-CPR ChannelRegistrationResponse message specifying Channel-ID and Push-Publication-Address, if appropriate.

When received DCD-3 ChannelSubscriptionRequest message that contains a reference to an unregistered channel, the DCD Server SHOULD issue DCD-CPR RequestForChannelRegistration (see Section 7.2.1.5) to the appropriate DCD Content Provider.

6.2.3.1 Emergency Channel Registration

DCD Service Providers can support DCD channels intended to deliver emergency alert services. DCD provides methods to define such services and ensure that they are always available as needed. The following requirements address those methods as they relate to channel registration, for DCD Service Providers that support DCD-based emergency alert services. Note that additional requirements may apply to broadcast bearer-based DCD as described in [DCD-TS-BCAST] and [DCD-TS-CBS].

These requirements ensure that DCD-based emergency alert services can always be provided, regardless of user or DCD-Enabled Client Application preferences or roaming status.

DCD Channels that are intended to carry emergency alert service information in an always-on manner SHALL be defined with the following channel metadata attributes:

- "suspend-allowed" set to "False"
- "deliver-when-roaming" set to "True"

DCD Servers SHOULD support auto-subscription to DCD Channels for emergency alert services, for all new users.

DCD Servers SHALL prevent unsubscription to DCD Channels for emergency alert services.

DCD Servers SHALL ensure that DCD Channels for emergency alert services cannot be suspended.

6.2.4 Channel Subscription

6.2.4.1 Internal Subscription

The following refers to DCD Server handling of the channel subscription request from the DCD Client.

Upon receiving an acceptable DCD-3 ChannelSubscriptionRequest message (see Section 7.1.3.7) from the DCD Client and if the notify-on-subscription-change Channel Metadata attribute (see Section 8.2) is set to "True" or the proxied-subscription Channel Metadata attribute is set to "False" for the subscribed channel:

- The DCD Server SHALL send the DCD-CPR SubscriptionRequest (see Section 7.2.1.1) to the DCD Content Provider. The DCD Server SHALL include in the DCD-CPR SubscriptionRequest the Subscriber-Info parameter as well as all applicable parameters from the DCD-3 ChannelSubscriptionRequest.

- The DCD Server SHALL send the DCD-CPR SubscriptionRequest to the Content Provider at the cp-subscription-manager-address, if specified in the channel metadata.

If the DCD-3 ChannelSubscriptionRequest message is not acceptable, e.g. fails some validation as described in Section 6.2.4.3, the DCD Server SHALL return an ErrorNotification to the DCD Client, with error details as appropriate. Upon a successful validation and successful DCD-CPR SubscriptionResponse (see Section 7.2.1.1) from the DCD Content Provider, if required, the DCD Server SHALL send a DCD-3 ChannelSubscriptionResponse (see Section 7.1.3.7) to the DCD Client. The DCD Server SHALL include in the Channel-Metadata all applicable metadata attributes.

If the DCD-3 ChannelSubscriptionRequest message contained a reference to an unregistered channel, the DCD Server SHOULD issue a DCD-CPR RequestForChannelRegistration message (see Section 7.2.1.5) to the appropriate DCD Content Provider. Request for channel registration by the DCD Server is subject to Service Provider policies (see Section 4.3) and to

technical feasibility (e.g. availability of access to Content Provider). After completion of channel registration (see Section 6.2.3), the DCD Server SHALL send a DCD-3 ChannelSubscriptionResponse as described above.

The DCD Server SHALL perform the procedures in Section 6.2.4.3 DCD Server Channel-Metadata Actions.

If the subscription request is not accepted, the DCD Server SHALL respond with an error.

If the DCD Server is delivering the subscribed channel over BCAST or CBS, the DCD Server SHALL also follow the rules and procedures of [DCD-TS-BCASTor DCD-TS-CBS] in addition to those listed above.

6.2.4.2 External Subscription

The following refers to DCD Server handling of the channel subscription notification from the DCD Content Provider.

Upon receiving a DCD-CPR SubscriptionNotification message (see Section 7.2.1.3) from the DCD Content Provider, the DCD Server SHALL send a DCD-3 ChannelSubscriptionNotification (see Section 7.1.3.9) to the DCD Client identified according to the Delivery-Endpoint-Info parameter in DCD-CPR SubscriptionNotification message. The DCD Server SHALL include in the DCD-3 ChannelSubscriptionNotification all applicable parameters from the DCD-CPR SubscriptionNotification. The DCD Server MAY include in the Channel-Metadata parameter additional metadata attributes (see Section 8.2) required for the DCD Client and the DCD Enabled Client Application.

Upon receiving an acceptable DCD-3 ChannelSubscriptionNotificationResponse message (see Section 7.1.3.9) from the DCD Client, the DCD Server SHALL send a DCD-CPR SubscriptionNotificationResponse (see Section 7.2.1.3) to the DCD Content Provider. The DCD Server SHALL include in the DCD-CPR SubscriptionNotificationResponse all applicable parameters from the DCD-3 ChannelSubscriptionNotificationResponse.

If the DCD-3 ChannelSubscriptionNotificationResponse message was not acceptable, e.g. fails some validation as described in Section 6.2.4.3, the DCD Server SHALL:

- return an ErrorNotification to the DCD Client, with error details as appropriate
- send a DCD-CPR SubscriptionNotificationResponse message, with Subscription-Declined set to “True”

If the DCD-3 ChannelSubscriptionNotificationResponse message contained the Subscription-Declined element set to “True”, the DCD Server SHALL send a DCD-CPR SubscriptionNotificationResponse message, with Subscription-Declined set to “True”.

The DCD Server SHALL perform the procedures in Section 6.2.4.3 DCD Server Channel-Metadata Actions prior to sending DCD-3 ChannelSubscriptionNotification.

In the case of successful subscription validation, the DCD Server SHALL store any new or updated Channel Metadata related to the established subscription.

If the DCD Server is delivering the subscribed channel over BCAST or CBS, the DCD Server SHALL also follow the rules and procedures of [DCD-TS-BCAST or DCD-TS-CBS] in addition to those listed above.

6.2.4.3 DCD Server Channel Metadata Actions

If a dcd-2-broadcast-profile is available the DCD Server MAY verify that the requesting terminal is capable of receiving data on the requested connection(s).

If applicable, the DCD Server MAY verify that per the channel metadata for the channel, the requesting application/device nd subscriber are compatible with the channel, e.g.:

- the application profile of the requesting application, e.g. as compared to the matching-applications of the channel, or channel-selection-metadata of the application profile to the mime-types and content-types in the metadata of the channel
- the subscriber‘s profile, e.g. as compared to the genre and parental-rating of the channel
- channel-personalization options, as compared to the personalization features applicable for the user or device

- delivery-preferences-metadata and delivery-personalization-metadata, e.g. network-preferences, dcd-interface, and deliver-when-roaming, as compared to the user's device capabilities and network service plan
- charging metadata, as compared to the user's service plan, e.g. charging rules
- the current device profile, e.g.
    - static (UAProf) device properties as compared to the mime-types nd adaptation-capability of the channel
    - dynamic device properties, e.g. available DCD Client-managed storage, as compared to the storage-reservation requirements for the channel

When genre, parental-rating and/or content-types attributes are specified in the Channel Metadata, the DCD Server SHOULD validate via the OMA Categorization Based Content Screening [OMA-CBCS] enabler that the channel is suitable according to subscriber preferences and Service Provider policy.

6.2.4.4 Subscription Update

The following refers to DCD Server handling of the subscription update from the DCD Content Provider.

Upon receiving a DCD-CPR SubscriptionUpdate message (see Section 7.2.1.4) from the DCD Content Provider, the DCD Server SHALL:

- add subscriber identified by the Subscriber-Info parameter to the channel subscribers group associated with the Subscription-ID parameter if the Action parameter has value "Add"
- remove subscriber identified by the Subscriber-Info parameter from the channel subscribers group associated with the Subscription-ID parameter if the Action parameter has value "Remove"

6.2.4.5 Channel Unsubscription

The following refers to DCD Server handling of the channel unsubscription transactions.

Upon receiving a DCD-3 ChannelUnsubscriptionRequest message (see Section 7.1.3.8) from the DCD Client and if the notify-on-subscription-change Channel Metadata attribute (see Section 8.2) is set to "True" for the unsubscribed channel the DCD Server SHALL send the DCD-CPR UnsubscriptionNotification (see Section 7.2.1.2) to the DCD Content Provider. The DCD Server SHALL include in the DCD-CPR UnsubscriptionNotification the Subscriber-Info parameter as well as all applicable parameters from the DCD-3 ChannelUnsubscriptionRequest.

Upon processing of DCD-3 ChannelUnsubscriptionRequest, the DCD Server SHALL send a DCD-3 ChannelUnsubscriptionResponse message (see Section 7.1.3.8) to the DCD Client, and terminate content delivery operations for this channel for the subscriber.

If the unsubscription request is not accepted the DCD Server SHALL respond with an error.

Upon receiving a DCD-CPR ChannelDeregistrationRequest (see Section 7.2.1.6) from the DCD Content Provider, the DCD Server SHALL send DCD-3 ChannelUnsubscriptionNotification (see Section 7.1.3.8.2) to the DCD Clients that have applications subscribed to the deregistered channel. Additionally, the DCD Server MAY send DCD-3 ChannelUnsubscriptionNotification to the DCD Client without prior DCD-CPR ChannelDeregistrationRequest (e.g. due to charging dispute with the subscriber).

The DCD Server MAY also include a Reason value to clarify the reason for unsubscription in DCD-3 ChannelUnsubscriptionNotification.

If the DCD Server is delivering the subscribed Channel over BCAST and/or CBS, the DCD Server SHALL also follow the rules and procedures of [DCD-TS-BCAST, DCD-TS-CBS] in addition to those listed above.

6.2.5 Channel Metadata Update

The following refers to DCD Server handling of the DCD-3 ChannelMetadataUpdate message that is sent to inform the DCD Client about the changes to the existing Channel Metadata, e.g. upon DCD-CPR ChannelRegistrationRequest.

When necessary, the DCD Server SHALL create a ChannelMetadataUpdate message according to Section 7.1.3.13 and send an update to all DCD Clients subscribing to the updated Channel.

6.2.6 Content Delivery

6.2.6.1 Receive-Only Client Case

For the ***receive-only client*** case:

DCD Server delivers content to DCD Client over DCD-2 interface.

6.2.6.2 Content Publication

The following refers to DCD Server handling of content publication by Content Providers, using pull or push methods across the DCD-CPDE interface.

DCD Servers SHALL use content publication methods for each channel as defined by the publication-methods channel metadata attribute.

DCD Servers SHALL request content updates for channels per the publication-schedule channel metadata attribute, as applicable for channels, unless prohibited from doing so at the scheduled time, e.g.:

- if the Content Provider has suspended content updates for a channel.

In creating DCD-CPDE ContentUpdateRequest messages, DCD Servers SHALL include the channel-id for the channel to be updated.

For pull-based publication, DCD Servers SHALL store content and content metadata received in DCD-CPDE ContentUpdateResponse messages, for later delivery to subscribers.

For push-based publication, DCD Servers SHALL store content and content metadata received in DCD-CPDE ContentUpdate messages, for later delivery to subscribers.

If the subscription-id attribute is included in a ContentUpdate message, DCD Servers SHALL associate the update with the group of subscribers with the same subscription-id.

For channels that have both “DCD-2” and “DCD-1” values in the dcd-interface channel metadata attribute, the DCD Server SHALL choose the delivery interface per:

- the content metadata for specific content items
- the delivery-personalization-metadata and current delivery context of the target users, i.e. either the specific users indicated by the deliver-to content metadata attribute, or all users subscribed to the channel

6.2.6.3 Content Update Request

The following refers to general DCD Server handling of the DCD-1 ContentUpdateRequest message, and as sent for content update of one or more channels.

DCD Servers SHALL accept the DCD-1 ContentUpdateRequest message if received with a valid session-id.

DCD Servers SHALL serve content update requests, unless prohibited from doing so at the scheduled time, e.g.:

- if content delivery for the channel has been suspended

DCD Servers SHALL respond with an ErrorNotification message with error-code “channel not subscribed” to content update requests for subscription-based channels to which the requesting user is not subscribed.

DCD Servers SHALL respond with an ErrorNotification message with error-code “content unavailable” to content update requests for channels that are not currently available, i.e. before the channel-availability-start or after the channel-availability-end channel metadata attributes.

If interaction with the DCD Content Provider is required for a specific Content Update Request, the DCD Server SHALL send a DCD-CPDE ContentUpdateRequest message, including:

- the channel-id for the channel to be updated
- the set of subscription-id associated to the channel, for the specific subscriber

DCD Servers SHALL send a DCD-1 ContentUpdateResponse for update of all channels requested, if updated content is available.

If the list of channel IDs, content IDs and content addresses are not included in the DCD-1 ContentUpdateRequest message, the DCD Server SHALL provide update for all channels subscribed by the device.

DCD Servers SHALL include in the DCD-1 ContentUpdateResponse all content items appropriate for delivery via the DCD-1 interface, which:

- have been published by the Content Provider, with no deliver-at content metadata attribute
- have a deliver-at content metadata attribute, for which the deliver-at time has passed

Further handling is described under Section 6.2.6.6 Content Update Response.

6.2.6.4 On-Demand Request for Specific Content Item

If the ContentUpdateRequest message includes a content-id or content-address attribute, DCD Servers SHALL send a ContentUpdateResponse for the specific content item if the content item is still available, i.e.:

- the content item is still available in the DCD Server storage
- the related channel is still available, i.e. between the channel-availability-start and channel-availability-end channel metadata attributes.

6.2.6.5 Content Retrieval via content-address

If the ContentUpdateRequest message is received at an address associated with a specific content item, the DCD Server SHALL send a ContentUpdateResponse with the content, if the content is still available.

Where possible (e.g. for content items stored at DCD Server and related to the subscribed channel), the DCD Server SHOULD validate that delivery of the content item is allowed per current subscriptions and return error message otherwise. There is no expectation that content addresses other than those directly included in content metadata for previously delivered content items will be validated. For example, DCD Server MAY validate that a content request is associated with aux-content-link sent previously.

6.2.6.6 Content Update Response

The following refers to general DCD Server handling of DCD-1 ContentUpdateResponse messages.

In creating ContentUpdateResponse messages, DCD Servers SHALL include:

- the session-id from the ContentUpdateRequest message
- a content-package element for each content item in the response, including
    - the content metadata attributes (see Section 8.3)
    - the content, if to be provided in the content update response

6.2.6.7 Content Update Push

For channels that have a "DCD-2" value in the dcd-interface channel metadata attribute, DCD Servers SHALL send the DCD-2 ContentUpdatePush message when push-based content delivery to one or multiple users is required, e.g.:

- per the channel metadata push-schedule, for content items that have no specific deliver-at content metadata attribute
- per the deliver-at content metadata attribute, e.g. immediately when the deliver-at content metadata attribute is set at or before the current time, or per the schedule for a deliver-at content metadata attribute set later than the current time

DCD Servers SHALL send the DCD-2 ContentUpdatePush message when possible for specific users, e.g. according to the applicable network-preferences attribute.

If the delivery-spread channel metadata attribute is specified for the channel, DCD Servers SHALL distribute point-to-point DCD-2 ContentUpdatePush messages over the time period specified.

If the deliver-when-roaming channel metadata attribute is "False", the DCD Server SHALL not send the ContentUpdatePush message to a specific user if the DCD Server is aware that the user is roaming.

DCD Servers SHALL split delivery of a set of content items into more than one ContentUpdatePush message, if necessary (e.g. if content size exceeds available storage on device), with the minimum level of granularity being a content item.

In creating ContentUpdatePush messages, DCD Servers SHALL include:

- the session-id for the target DCD Client, if applicable (i.e. point-to-point push)
- a content-package element containing content and content metadata for each content item to be delivered in the message

If the network-preferences channel metadata attribute is specified in channel or content metadata, the DCD Server SHALL send the ContentUpdatePush message as specified in this attribute.

The DCD Server SHALL choose a bearer as appropriate for the specific channel or content item for delivery of the ContentUpdatePush message, e.g. OMA Push, Cell Broadcast, BCAST, etc.

In the case that Cell Broadcast or BCAST are used, the DCD Server SHALL comply with [DCD-TS-BCAST] or [DCD-TS-CBS] as applicable.

6.2.6.8 Content Update Notification

DCD Servers SHALL send the DCD-2 ContentUpdateNotification message when push-based content delivery to one or multiple users is required as described in Section 6.2.6.7, and pushing the entire content package is not feasible or desirable.

DCD Servers SHALL NOT use the DCD-2 ContentUpdateNotification message if use of the DCD-1 interface is not allowed for the channel (i.e. the dcd-interface channel metadata attribute does not contain a "DCD-1" value), since use of DCD-2 ContentUpdateNotification requires DCD Client ability to retrieve the referenced content via the DCD-1 interface.

If the deliver-when-roaming channel metadata attribute is "False", the DCD Server SHALL NOT send the ContentUpdateNotification message to a specific user if the DCD Server is aware that the user is roaming.

In creating the DCD-2 ContentUpdateNotification messages, DCD Servers SHALL include:

- the session-id for the target DCD Client, if applicable (i.e. point-to-point push)
- a new message-id value
- the applicable content metadata or DCD-Content Reference

6.2.6.9 Content Item Handling

The following refers to general DCD Server handling of content items.

If the delivery-priority content metadata attribute is set for a content item, the DCD Server SHALL expedite content delivery, e.g. immediately deliver the content item via the DCD-2 interface if appropriate.

If a DCD Channel is defined to carry both normal service alerts (e.g. service outage notifications) and emergency alert services, the DCD Server SHALL reserve the content metadata attribute "emergency-content" value "True" for exclusive use in actual emergencies.

DCD Servers SHOULD compress content in the most optimized form compatible with the DCD Client to which the content is being delivered.

DCD Servers SHALL set the content-encoding content metadata attribute per the applied content compression, if any.

If the replaces-content-id content metadata attribute is set for a content item, the DCD Server SHALL replace the indicated content item if still available in DCD Server storage. If the DCD Client requests a content item that has been replaced according to the replaces-content-id attribute of a newer content item, the DCD Server SHALL return the content item that replaced the requested content item.

If the content-expiration content metadata attribute is set for a content item, the DCD Server SHALL NOT deliver this content item after expiration. If the DCD Client requests an expired content item that has not been replaced by a newer content item, the DCD Server SHALL send an Error Notification message with error-code "content unavailable".

In providing pull-based or push-based content delivery via point-to-point bearers, DCD Servers SHALL customize and personalize the content, e.g. as required by:

- the channel-metadata for associated channels, e.g. channel-personalization options
- the content-metadata for available content items, e.g. deliver-per-location, deliver-per-presence, deliver-per-xdms, network-preferences, deliver-when-roaming, parental-rating, delivery-priority, content-length, content-types
- the current device profile, e.g. static (UAProf) and dynamic device properties, available DCD Client-managed storage

When processing the content item, the DCD Server SHALL apply Content Metadata attributes (as described above), and SHALL apply Channel Metadata attributes specified for the channel, unless Channel Metadata attributes conflict with corresponding Content Metadata attributes, in which case the DCD Server SHALL use the latter (i.e. Content Metadata attributes override corresponding Channel Metadata attributes).

For content items being delivered via progressive download, the DCD Server SHALL set the aux-content-link metadata attribute to the URI of the next download segment, if any.

When parental-rating and/or content-types attributes are specified in the Content Metadata, the DCD Server MAY support content filtering via the OMA Categorization Based Content Screening [OMA-CBCS] enabler.

DCD Server SHALL follow the packaging rules as defined in Section 9.1.

6.2.6.10 Content Confirmation

If the content-delivery-notification content metadata attribute is set for a content item, the DCD Server SHALL expect DCD-3 ContentDeliveryConfirmation message following DCD-3 ContentUpdateResponse or ContentUpdatePush.

When received DCD-3 ContentDeliveryConfirmation, the DCD Server SHOULD either:

- resend content item if delivery status for this content item is specified as "False"
- send Error Message (see Section 13) to the DCD Content Provider for the content item with delivery status "False" (e.g. if repeat failure for this content item)

6.2.7 Content Submission

Upon reception of the DCD-1 ContentSubmitRequest message (see Section 7.1.1.2), DCD Servers SHALL send the DCD-CPDE ContentUpdateRequest message to the DCD Content Provider at the submit-address if indicated, or the pull-publication-address for the channel if the submit-address is not indicated. The DCD Server SHALL include in the

ContentUpdateRequest message the Submit-Package parameter from the ContentSubmitRequest message and other parameters as described in Section 7.2.2.2).

Upon reception of DCD-CPDE ContentUpdateResponse message (see Section 7.2.2.2), the DCD Server SHALL send the ContentUpdateResponse message (see Section 7.1.1.1) to the DCD Client if new content needs to be sent to the DCD Client for this channel (e.g. Content-Package parameter is present in DCD-CPDE ContentUpdateResponse).

If new content is not available for this channel, the DCD Server SHALL send ContentSubmitConfirmation message (see Section 7.1.1.2) to the DCD Client.

6.2.8 Channel Suspension

Suspend/resume transactions SHALL NOT apply to "emergency" channels as defined in [Section 6.2.3.1Emergency Channel Registration].

6.2.8.1 Suspension Effect Upon DCD Transactions

When all non-"emergency" channels are suspended for the DCD Client, the DCD Server SHALL NOT send the DCD-3 ChannelDiscoveryInfo message or DCD-3 ChannelDiscoveryNotification message to the DCD Client via point-to-point Push bearers.

When a channel is suspended, the DCD Server SHALL NOT send DCD-2 ContentUpdatePush message or ContentUpdateNotification message to the DCD Client via point-to-point Push bearers.

If a content delivery transaction is in progress upon reception of a suspension request or detection of a condition requiring suspension, the DCD Server SHALL either terminate or continue with delivery of the content based upon DCD Service Provider policy.

If the DCD Server terminates the content delivery it SHALL mark the transaction as incomplete based upon DCD Service Provider policy.

When a previously suspended DCD Channel is resumed, the DCD Server SHALL deliver any buffered content that has not yet been delivered to applicable DCD Clients.

6.2.8.2 Initiated by the DCD Client

Upon the DCD-3 ChannelSuspendRequest message from the DCD Client, the DCD Server SHALL send the DCD-3 ChannelSuspendResponse message to the DCD Client.

If the Content Provider is handling charging for one or more affected channels, and therefore has to be notified of channel suspension, the DCD Server SHALL send the DCD-CPDE ChannelSuspendRequest message to each Content Provider for its affected channel(s), with the subscriber information before sending DCD-3 ChannelSuspendResponse. Upon reception of the DCD-CPDE ChannelSuspendResponse message, the DCD Server SHALL send the DCD-3 ChannelSuspendResponse message to the DCD Client.

After the DCD Client initiated suspension is completed, only the DCD Client can resume content delivery.

Upon the DCD-3 ChannelResumeRequest message from the DCD Client, the DCD Server SHALL:

- send the DCD-3 ChannelResumeResponse message to the DCD Client
- resume DCD operations for the affected channel(s) for the subscriber.

If the Content Provider is handling charging for one or more affected channels, and therefore has to be notified of channel resumption, the DCD Server SHALL send the DCD-CPDE ChannelResumeRequest message to each Content Provider for its affected channel(s), with the subscriber information before sending DCD-3 ChannelResumeResponse. Upon reception of the DCD-CPDE ChannelResumeResponse message, the DCD Server SHALL send the DCD-3 ChannelResumeResponse message to the DCD Client.

6.2.8.3 Initiated by the DCD Server

If the DCD Server wants to suspend content delivery for one or several channels to a specific DCD Client (e.g. upon roaming, for network maintenance, or a charging problem), it SHALL send the DCD-3 ChannelSuspendNotification message to the DCD Client with the target Channel IDs.

If the Content Provider is handling charging for one or more affected channels, and therefore has to be notified of channel suspension, the DCD Server SHALL send the DCD-CPDE ChannelSuspendRequest message to each Content Provider for its affected channel(s) , with the subscriber information when appropriate e.g. suspension upon roaming.

After the DCD Server initiated suspension is completed, only the DCD Server can resume content delivery.

When the DCD Server needs to resume content delivery, the DCD Server SHALL:

- send the DCD-3 ChannelResumeNotification to the DCD Client with the target Channel IDs
- resume DCD operations for the affected channel(s).

If the Content Provider is handling charging for one or more affected channels, and therefore has to be notified of channel resumption, the DCD Server SHALL send the DCD-CPDE ChannelResumeRequest message to each Content Provider for its affected channel(s) , with the subscriber information when appropriate.

6.2.8.4 Initiated by the Content Provider

Upon the DCD-CPDE ChannelSuspendNotification message from a Content Provider, the DCD Server SHALL send the DCD-3 ChannelSuspendNotification message to all subscribed DCD Clients.

Upon the DCD-CPDE ChannelResumeNotification message from a Content Provider, the DCD Server SHALL:

- send the DCD-CPDE ChannelResumeConfirmation message to the Content Provider
- send the DCD-3 ChannelResumeNotification message to all subscribed DCD Clients

6.2.9 Channel Discovery

6.2.9.1 Channel Guide creation

The DCD Server SHALL ensure that channels offered in the channel guide are in compliance with the Application Profile of the targeted DCD Enabled Client Application.

The DCD Server SHALL include only the Channel Metadata subset applicable to the DCD Enabled Client Application [Section 8.2] when sending Channel Discovery Info.

6.2.9.2 Channel Discovery Request from the DCD Client

Upon receiving the DCD-3 ChannelDiscoveryRequest message the DCD Server SHALL send the Channel Discovery Info message to the requesting DCD-Enabled Client Application via the DCD Client.

6.2.9.3 Channel Discovery Info from the DCD Server

The DCD Server SHALL send the DCD-3 ChannelDiscoveryInfo message to the DCD-Enabled Client Application with at least one action:

1. updating the DCD Enabled Client Application with changes to the channels registered at the DCD Server (new, updated, and/or removed channels).
2. replacing the Channel Guide on the DCD Enabled Client Application via the Channel-removed attribute with "*" and Channel-added attribute with the list of all matching channels

6.2.9.4 Channel Discovery Notification

DCD Server MAY send DCD-3 ChannelDiscoveryNotification message to notify the DCD Enabled Client Application about the changes to the channel guide related to this application.

6.2.10 Usage Tracking Report

The DCD Server receives a UsageTrackingReport message from the DCD Client reporting its usage tracking statistical information. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive only terminal).

The DCD Server SHALL process UsageTrackingReport messages from the DCD Client. The messages may be scheduled, or upon an event, or upon DCD Server's request. Refer to Section 6.1.9 for details on the conditions for DCD Client behavior.

The DCD Server MAY send a RequestForUsageTrackingReport message to the DCD Client if it requires an update of usage tracking statistical information.

The RequestForUsageTrackingReport message SHALL be created with the attributes as defined in Section 7.1.3.6.1.

If all DCD Channels to a particular DCD Client are suspended, the DCD Server SHALL not send RequestForUsageTrackingReport messages until at least one DCD Channel is resumed.

6.2.11 Contextual Information Upload

The DCD Server receives a ContextualInformationUpload message from the DCD Client reporting on its contextual information. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive only terminal).

The DCD Server SHALL process ContextualInformationUpload messages from the DCD Client. The messages may be scheduled or upon an event. Refer to Section 6.1.10 for details on the conditions for DCD Client behavior.

The DCD Server SHALL utilize Contextual Information Data when delivering the DCD Content.

The DCD Server MAY send a ContextualInformationUploadRequest message to the DCD Client if it requires an update of contextual information.

The ContextualInformationUploadRequest message SHALL be created with the attributes as defined in Section 7.1.3.5.1

If all DCD Channels to a particular DCD Client are suspended, the DCD Server SHALL not send ContextualInformationUploadRequest messages until at least one DCD Channel is resumed.

6.2.12 Charging

The following refers to DCD Server handling of charging as described in [DCD-TS-Charging]

When the DCD Server registers a new channel with a Content Provider, it SHALL check in the 'Charging-rules' channel metadata who is the entity in charge of the charging.

If the DCD Server is responsible for handling the charging, it SHALL specify in the Channel Guide the purchase options which define each charging scenario and the associated price. If the DCD Server does not define any purchase options in the channel metadata, the channel SHALL be seen as free of charge.

If the charging is handled by the Content Provider, the DCD Server SHALL forward the purchase option in the Channel Guide if received during channel registration.

When the DCD server receives a DCD-CPR SubscriptionNotification message from a DCD Content Provider, the DCD Server SHALL forward the selected purchase option in the SubscriptionNotification message to the DCD Client if present in the DCD-CPR SubscriptionNotification message.

If the DCD Server receives a DCD-3 ChannelSubscriptionRequest message or a DCD-CPR SubscriptionNotificationResponse with an outdated purchase-option-id, it SHALL respond with an ErrorNotification message with error-code "purchase option outdated" to the subscription.

If the Content Provider requested to be notified for its channel subscriptions and provided purchase option metadata, the DCD Server shall specify the Selected-Purchase-Option-Id parameter in the DCD-CPR SubscriptionRequest message.

The DCD Server SHALL uniquely identify each charging option with the purchase-option-id. If the price associated to a purchase option is modified, the DCD Server SHALL also modify the purchase-option-id to ensure that when a DCD Client subscribes to a channel, it is charged for the chosen purchase option and the announced price.

6.2.13 Content Repair

Upon reception of the DCD-3 ContentRepairRequest message from the DCD Client, the DCD Server SHALL send the DCD-3 ContentRepairResponse message to the DCD Client as described in Section 7.1.3.12. The DCD Server SHALL include the requested content package for repair in the DCD-3 ContentRepairResponse message as either the actual content or the reference to the link to the content.

7. DCD Interfaces

7.1 Client-Server Interfaces

Implementation requirements stated in each of the following subsections only apply for both the DCD Client and DCD Server, unless stated otherwise.

7.1.1 Interface DCD-1

7.1.1.1 Content Update

The Content Update is described in Figure 15: Content Update. The transaction allows the DCD Client to request content available for download. The Content Update transaction consists of the following messages:

- ContentUpdateRequest from the DCD Client to the DCD Server;
- ContentUpdateResponse from the DCD Server to the DCD Client;
- ContentDeliveryConfirmation from the DCD Client to the DCD Server

The DCD Client sends a ContentUpdateRequest to the DCD Server when one or more of the following events occur:

- Content Update Notification from the DCD Server
- On demand content update request (i.e. content update without notification from DCD Server, triggered by a DCD-Enabled Client Application, scheduled content update, content expiration etc.).

ContentDeliveryConfirmation: the DCD Client sends ContentDeliveryConfirmation message to the DCD Server if content confirmation is requested by the DCD Server. The confirmation is conditional upon:

- availability of the uplink from DCD Client to the DCD Server (e.g. in receive-only terminal)
- request for confirmation as specified in Channel Metadata or Content Metadata

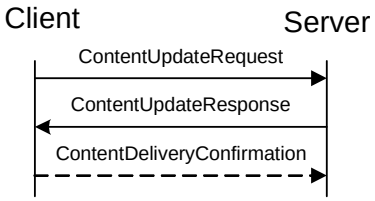


Figure 15: Content Update Transaction

7.1.1.1.1 Message and Information Elements

Message	Implementation	Direction
ContentUpdateRequest	Mandatory	DCD Client → DCD Server
ContentUpdateResponse	Mandatory	DCD Client ←DCD Server
ContentDeliveryConfirmation	Mandatory	DCD Client → DCD Server

Table 2 Message Directions for Content Update between DCD Client and DCD Server

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within

			the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-IDs	Optional	String	Comma separated list of IDs for channels to be updated.
Content-IDs	Optional	String	Comma separated list of Content-IDs of the specific content items requested.
Content-addresses	Optional	String	Comma separated list of Content-addresses of the specific content items requested.

Table 3 Information elements in ContentUpdateRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-ID	Optional	String	ID of the channel related to the content package. If absent, the content metadata in the Content-Package element SHALL contain Channel-ID attribute.
Content-Package	Mandatory	List of Data Structures	One or more content packages. The content package consists of content payload and content metadata. Content payload could be empty if content-address parameter of the content metadata contains the address where the content could be retrieved (e.g. URI). If Channel-ID element is present in the message all items in Content-Package element belong to the same channel (identified by Channel-ID element).

Table 4 Information elements in ContentUpdateResponse message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Content-Confirmation	Mandatory	List	List of pairs: Content Identifier and status set by the DCD Client. Status values: { True | False }

Table 5 Information elements in ContentDeliveryConfirmation message

7.1.1.2 Content Submission

The Content Submission transaction is illustrated in Figure 16: Content Submission Transaction. This transaction is initiated by DCD Clients upon request of DCD-Enabled Client Applications. The Content Submission transaction may or may not result in the DCD Content returned to the DCD Client in response. If there is DCD Content returned in response to content submission, the transaction consists of the following messages:

- ContentSubmitRequest from the DCD Client to the DCD Server
- ContentUpdateResponse (see Section 7.1.1.1) from the DCD Server to the DCD Client
- ContentDeliveryConfirmation (see Section 7.1.1.1) from the DCD Client to the DCD Server

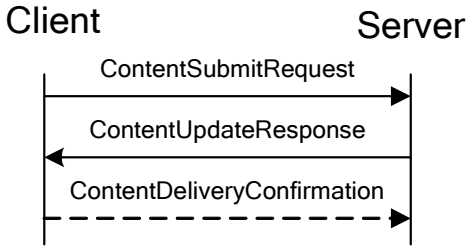


Figure 16: Content Submission Transaction with DCD Content returned to the DCD Client

If there is no DCD Content returned in response to content submission, the transaction consists of the following messages:

- ContentSubmitRequest from the DCD Client to the DCD Server
- ContentSubmitConfirmation from the DCD Server to the DCD Client

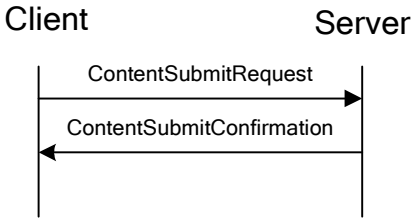


Figure 17: Content Submission Transaction without DCD Content returned to the DCD Client

7.1.1.2.1 Message and Information Elements

Message	Implementation	Direction
ContentSubmitRequest	Mandatory	DCD Client → DCD Server
ContentUpdateResponse (see Section 7.1.1.1)	Mandatory	DCD Client ←DCD Server
ContentDeliveryConfirmation (see Section 7.1.1.1)	Mandatory	DCD Client → DCD Server
ContentSubmitConfirmation	Mandatory	DCD Client ←DCD Server

Table 6 Message Directions for Content Submission between DCD Client and DCD Server

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-ID	Mandatory	String	ID of channel related to the submitted content.
Submit-Address	Conditional	String	URI to which the enabler should deliver the submitted content. SHALL be present if specified in the DCD-CADE ContentSubmitRequest.
Submit-Package	Mandatory	Opaque Data	Content to be submitted. Submit-Package can contain user identity and/or other metadata that is targeted for the Content Provider and opaque for the DCD Enabler.

Table 7 Information elements in ContentSubmitRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 8 Information elements in ContentSubmitConfirmation message

7.1.2 Interface DCD-2

7.1.2.1 Content Update Push

The Content Update Push transaction is shown in Figure 18: Content Update Push transaction. The transaction allows DCD Server to send content package to the DCD Client, when updated DCD Content is available. The Content Update Push transaction consists of the following messages:

- ContentUpdatePush from the DCD Server to the DCD Client;
- ContentDeliveryConfirmation from the DCD Client to the DCD Server

ContentUpdatePush: the DCD Server sends ContentUpdatePush message to the DCD Client to provide updated DCD Content. The message is sent over push or broadcast bearer. If the message is not valid, the DCD Client responds with the appropriate error message (refer to Section 13).

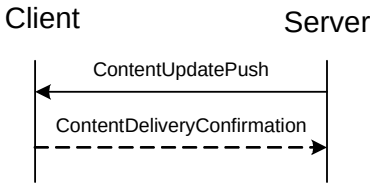


Figure 18: Content Update Push transaction

7.1.2.1.1 Message and Information Elements

Message	Implementation	Direction
ContentUpdatePush	Mandatory	DCD Client ← DCD Server
ContentDeliveryConfirmation (see Section 7.1.1.1.1)	Mandatory	DCD Client → DCD Server

Table 9 Message directions for Content Push transaction

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Conditional	String	Identifies this message. The Message-ID is unique within a current session. Presence in the message is conditional upon the session-id being present. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-ID	Optional	String	ID of the channel related to the content package. If absent, the content metadata in the Content-Package element SHALL contain Channel-ID attribute.
Content-Package	Mandatory	List of Data Structures	One or more content packages. The content package consists of content payload and content metadata. If Channel-ID element is present in the message all items in Content-Package element belong to the same channel (identified by Channel-ID element).

Table 10 Information elements in ContentUpdatePush Message

7.1.2.2 Content Update Notification

The ContentUpdateNotification message triggers one or more Content Update transactions as shown in Figure 19: Content Update Notification Initiated Content Update transaction. This message allows DCD Server to send to the DCD Client notification about DCD Content availability. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal). The Content Update Notification initiated Content Update transaction consists of the following messages:

- ContentUpdateNotification from the DCD Server to the DCD Client;
- ContentUpdateRequest from the DCD Client to the DCD Server (see Section 7.1.1.1 for details)
- ContentUpdateResponse from the DCD Server to the DCD Client (see Section 7.1.1.1 for details);
- ContentDeliveryConfirmation from the DCD Client to the DCD Server (see Section 7.1.1.1 for details)

ContentUpdateNotification: the DCD Server sends ContentUpdateNotification message to the DCD Client to notify about availability and additional charging information of updated DCD Content including the content metadata e.g. content-price. The message is sent over push or broadcast bearer.

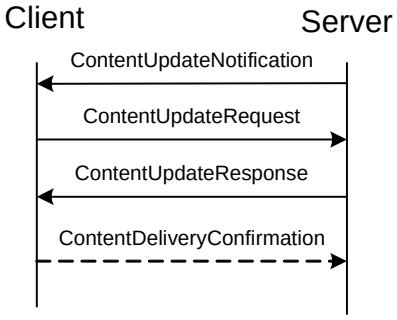


Figure 19: Content Update Notification Initiated Content Update transaction

7.1.2.2.1 Message and Information Elements

Message	Implementation	Direction
ContentUpdateNotification	Mandatory	DCD Client ←DCD Server
ContentUpdateRequest (see Section 7.1.1.1)	Mandatory	DCD Client → DCD Server
ContentUpdateResponse (see Section 7.1.1.1)	Mandatory	DCD Client ←DCD Server
ContentDeliveryConfirmation (see Section 7.1.1.1)	Mandatory	DCD Client → DCD Server

Table 11 Message directions for Content Notification Initiated Content Update transaction

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name.

			This parameter is present when CBS bearer is used for this transaction.
Message-ID	Conditional	String	Identifies this message. The Message-ID is unique within a current session. Presence in the message is conditional upon the session-id being present. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
DCD-Content-Reference	Conditional	String	Reference to the DCD Content and associated content metadata. Present if Content-Metadata is not present.
Content-Metadata	Conditional	List of Structures	Subset of DCD Content metadata for available content items, e.g. content-price for advice of charge. Present if DCD-Content-Reference is not present.

Table 12 Information elements in ContentUpdateNotification Message

7.1.3 Interface DCD-3

7.1.3.1 Client Activation

The Client Activation activates the DCD service for a specific DCD Client. During the activation, the subscriber authentication is performed separately according to the security level and authentication model described in Section 10.1 and 10.2.

This section gives the details of the transaction messages.

Authentication is an optional step, the use of which depends upon policies of the DCD Server. Depending upon the authentication method required by the DCD Server, authentication may not involve the DCD Client directly. “Figure 20: Client Activation without Authentication by DCD Client” shows client activation when no authentication is required by the DCD Server (e.g. it may already have occurred, or not be required by the DCD Server), or the authentication method is transparent to the DCD Client.

Device capabilities can be accessed by the DCD Server based upon the device ID provided in the ClientActivationRequest message, and the UAProf referred to in the HTTP request.

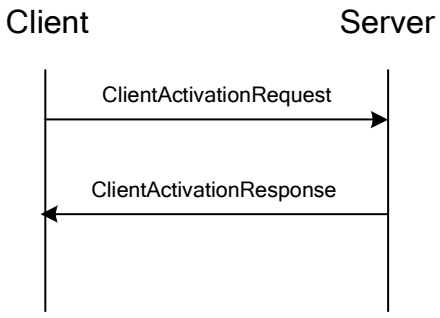


Figure 20: Client Activation without Authentication by DCD Client

“Figure 21: Client Activation with Authentication by DCD Client” shows client activation with authentication required by the DCD Server.

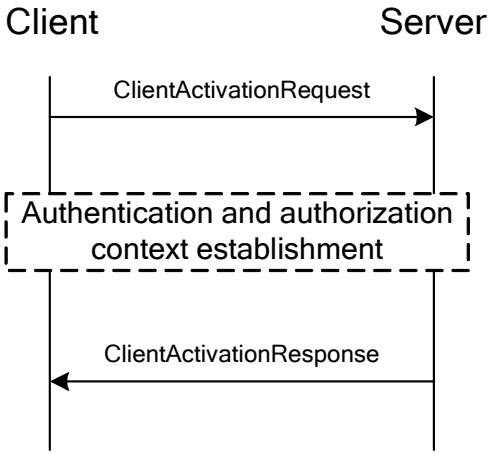


Figure 21: Client Activation with Authentication by DCD Client

7.1.3.1.1 Message and Information Elements

Message	Implementation	Direction
ClientActivationRequest	Mandatory	Client → Server
ClientActivationResponse	Mandatory	Client ← Server

Table 13 Message directions for Client Activation

Information Element	Req	Type	Description
Device-ID	Optional	String	Device identifier, e.g. device make / model and version of the DCD Client.
Version	Mandatory	String	The highest version of protocol supported in the device. The format of the version of the DCD enabler is defined as follows: <major version>.<minor version> The current version of the DCD enabler is "1.0".

Table 14 Information elements in ClientActivationRequest message

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the subscriber. It is present only if CBS technology is supported by the target DCD Client and the DCD Service provider. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name.
Session-TTL	Optional	Integer	The server required TTL in seconds for the session. '0' means an infinite session.

Table 15 Information elements in ClientActivationResponse message

7.1.3.1.2 Client Activation request by a DCD Server

The DCD Server may request Client Activation upon commencement of a DCD service or following the event of deactivation of the DCD Client by the DCD Server. In the latter case, in order to renew the DCD service the DCD Server SHALL issue activation request to the DCD Client (i.e. the activation request of the DCD Client follows the request for Client Activation from the DCD Server).

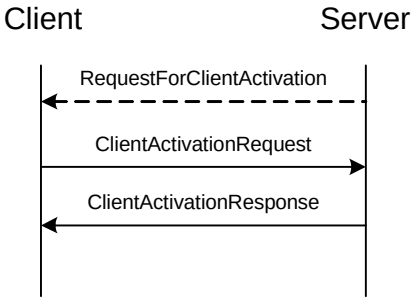


Figure 22: Client Activation by request from the DCD Server

7.1.3.1.2.1. Message and Information Elements

Message	Implementation	Direction
RequestForClientActivation	Optional	Client ← Server
ClientActivationRequest (see 7.1.3.1.1)	Mandatory	Client → Server
ClientActivationResponse (see 7.1.3.1.1)	Mandatory	Client ← Server

Table 16 Message directions for Client Activation by request from the DCD Server

Information Element	Req	Type	Description
dcd-3-connection-profile-name	Optional	String	Name of a locally configured DCD-3 connection profile. If not present, the message applies to the default dcd-3-connection-profile
dcd-3-connection-profile	Optional	Structure	Specific DCD-3 connection profile to use.

Table 17 Information elements in RequestForClientActivation message

The messages ClientActivationRequest and ClientActivationResponse are as described in Section 7.1.3.1.1 and represent authenticated or non-authenticated client activation process.

7.1.3.2 Client Deactivation

Upon deregistration of the last DCD-Enabled Client Application, the DCD Client could deactivate with the DCD Server. That means the current session is no longer needed, since there are no DCD Enabled Client Applications being served. To deactivate, the DCD Client sends a ClientDeactivationRequest to the DCD Server. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

After this request:

- The indicated session is removed
- The DCD Server can respond with a failure to deactivate. After the failure response, the DCD Client could retry the deactivation.

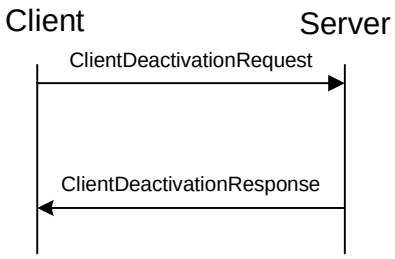


Figure 23: Client Deactivation Request

7.1.3.2.1 Message and Information Elements

Message	Implementation	Direction
ClientDeactivationRequest	Mandatory	Client → Server
ClientDeactivationResponse	Mandatory	Client ← Server

Table 18 Message directions for Client Deactivation

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session established in the Client Activation. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 19 Information elements in ClientDeactivationRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session established in the Client Activation. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 20 Information elements in ClientDeactivationResponse message

7.1.3.2.2 Client Deactivation by DCD Server

The DCD Server deactivates the DCD Client in order to disable the DCD Service. Upon deactivation the DCD Clients will no longer have sessions with this DCD Server. Deactivation by the DCD Server does not require confirmation by the DCD Client.

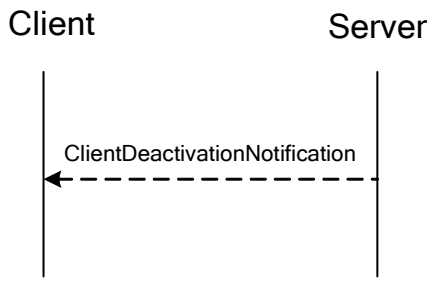


Figure 24: Client Deactivation by the DCD Server

7.1.3.2.2.1. Message and Information Elements

Message	Implementation	Direction
ClientDeactivationNotification	Optional	Client ← Server

Table 21 Message directions for Client Deactivation by the DCD Server

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier of the current session established in the Client Activation. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Reason	Optional	String	Free text describing the reason for deactivation

Table 22 Information elements in ClientDeactivationNotification message

7.1.3.3 Application Registration

The Application Registration process involves registration of a DCD-Enabled Client Application with the DCD Server or re-registration of a DCD-Enabled Client Application by providing an updated Application Profile. The prerequisite to registration is DCD Client activation with the DCD Server. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

This section gives the details of the transaction messages.

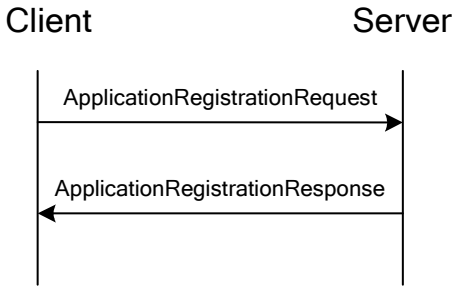


Figure 25: Application Registration

7.1.3.3.1 Message and Information Elements

Message	Implementation	Direction
ApplicationRegistrationRequest	Mandatory	Client → Server
ApplicationRegistrationResponse	Mandatory	Client ← Server

Table 23 Message directions for Application Registration

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session established in the Client Activation. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Application-Profile	Mandatory	Structure	Contains part of the application profile relevant to the DCD Server: Application-ID and Channel-Selection-Metadata. The details of the Application-Profile are defined in Section 8.1.

Table 24 Information elements in ApplicationRegistrationRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-Guide	Mandatory	List	The list of channel metadata structures for the channels matching application profile preferences. Each element in the list represents the metadata for a single channel. The channel metadata in the channel guide contains the subset of channel metadata parameters relevant to the DCD Enabled Client Application (see Sections 8.2 and 13.4 for details).

Table 25 Information elements in ApplicationRegistrationResponse message

7.1.3.4 Application Deregistration Notification

This transaction allows the DCD Client to notify the DCD Server of DCD-Enabled Client Application deregistration. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

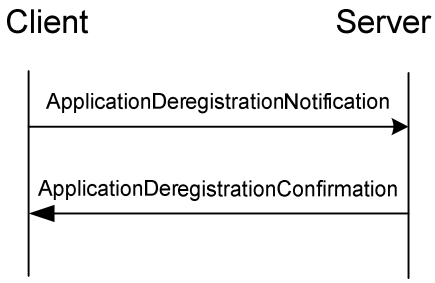


Figure 26: Application Deregistration Notification

7.1.3.4.1 Message and Information Elements

Message	Implementation	Direction
ApplicationDeregistrationNotification	Mandatory	Client → Server
ApplicationDeregistrationConfirmation	Mandatory	Client ← Server

Table 26 Message directions for Application Deregistration Notification

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session established in the Client Activation. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.

Table 27 Information elements in ApplicationDeregistrationNotification message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 28 Information elements in ApplicationDeregistrationConfirmation message

7.1.3.5 Contextual Information Upload

The contextual information upload transaction (see Figure 27: Contextual Information Upload) occurs when the contextual information on the client device has changed, upon predefined schedule, or if requested by the DCD Server. The contextual information includes the capabilities (e.g. available storage), device status (e.g. roaming status and PLMN code), etc. Via the ContextualInformationUploadRequest message, the DCD Server can both request an immediate contextual information report, and set policies controlling when the DCD Client should automatically report contextual information.

This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

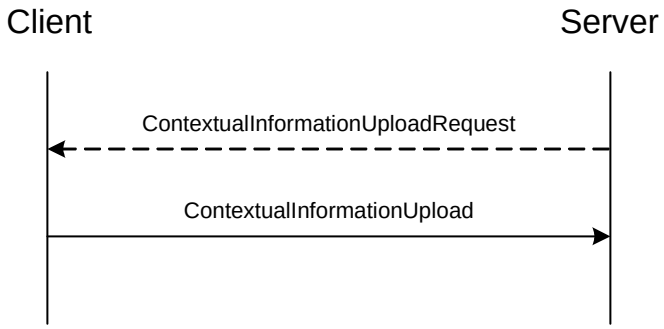


Figure 27: Contextual Information Upload

7.1.3.5.1 Message and Information Elements

Message	Implementation	Direction
ContextualInformationUploadRequest	Optional	Client ← Server
ContextualInformationUpload	Mandatory	Client → Server

Table 29 Message directions in Contextual Information Upload transaction

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. The Session-ID present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Conditional	String	Identifies this message. The Message-ID is unique within a current session. Presence in the message is conditional upon the session-id being present. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Report-policy	Optional	Structure	Set of policies for automatic reporting of contextual information. If present, the DCD Client SHALL use

			the indicated policies for automatic reporting of contextual information. If not present, the DCD Client SHALL treat this as an on-demand request for Contextual Information Upload. This attribute will replace the prior policy, if set. If the value of any attribute in the policy is empty, the DCD Client SHALL remove the related policy setting. If the report-policy is empty (all options are absent), the DCD Client SHALL remove all policy settings and take no further action.

Table 30 Information elements in ContextualInformationUploadRequest Message

Information Element	Req	Type	Description
report-frequency	Optional	String	Interval (in minutes) between automatic reports. If present, the DCD Client SHALL send a ContextualInformationUpload message at this frequency, beginning at the time the ContextualInformationUploadRequest was received. The default value is "off", i.e. there is no schedule reporting.
Storage-report-policy	Optional	List of Structures	List of policies for reporting on reserved and free storage managed by the DCD Client.
Roaming-status-report	Optional	String	Policy for reporting on roaming, one of "on-demand" (default value), "any-change", "home", "roam", "international". If "home", "roam", or "international", the DCD Client SHALL report only upon a change to/from the specified roaming status.
available-bearers-report	Optional	String	Policy for reporting on available bearers, one of "on-demand" (default value), "any-change", "EDGE", "UMTS", "CDMA2000", "GAN", "WiMAX", "LTE", "802.11", "CBS", "BCAST". If "EDGE", "UMTS", "CDMA2000", "GAN", "WiMAX", "LTE", "802.11", "CBS", "BCAST", the DCD Client SHALL report only upon a change of availability for the specified bearer.
preferred-bearers-report	Optional	String	Policy for reporting on preferred bearers, one of "on-demand" (default value), "any-change".

Table 31 Information elements in Report-policy structure

Information Element	Req	Type	Description
Channel-ID	Mandatory	String	ID of channel to be reported upon. If empty, applies to all channels (i.e. total of DCD Client managed storage).

Reserved-storage-report	Optional	String	Policy for reporting on reserved storage (for the specified channel, if applicable) managed by the DCD Client, one of "on-demand" (default value), "any-change".
Free-storage-report	Optional	String	Policy for reporting on free storage (within the reserved storage), one of "on-demand" (default value), "exhausted". If "exhausted", the DCD Client SHALL send the ContextualInformationUpload message when no storage is available for received content.

Table 32 Information elements in Storage-report-policy structure

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Contextual-Information	Mandatory	Structure	The dynamic contextual information of the device.

Table 33 Information elements in ContextualInformationUpload message

Information Element	Req	Type	Description
Storage-report	Conditional	List of Structures	Storage status for the set of channels being reported upon. Present for on-demand reports. Present for reports triggered by an applicable report-policy. Conditional upon storage reservation being implemented.
roaming-status	Conditional	String	Current roaming status, one of "unknown", "home", "roam", "international". Present for on-demand reports, or reports triggered by an applicable report-policy.
available-bearers	Conditional	String	Types of all available bearer networks: comma-separated list of one or more of "unknown", "EDGE", "UMTS", "CDMA2000", "GAN", "WiMAX", "LTE", "802.11",, "CBS", "BCAST". Present for on-demand reports, or reports triggered by an applicable report-policy.
preferred-bearers	Conditional	String	Preference for specific bearers, if available, in priority order: comma-separated list of one or more of "unknown", "EDGE", "UMTS", "CDMA2000", "GAN", "WiMAX", "LTE", "802.11",, "CBS", "BCAST". Present for on-demand reports, or reports triggered by an applicable report-policy.

Table 34 Information elements in ContextualInformation structure

Information Element	Req	Type	Description
Channel-ID	Mandatory	String	ID of channel being reported upon. If set to "*", applies to all channels (i.e. total of DCD Client managed storage).
Reserved-storage	Conditional	Integer	Memory in bytes that is currently reserved (for the specified channel, if applicable) for the channel in DCD content storage managed by the DCD Client. Present for on-demand reports, or reports triggered by an applicable report-policy.
Free-storage	Conditional	Integer	Amount of free memory in bytes in the reserved DCD content storage for the channel. Present for on-demand reports, or reports triggered by an applicable report-policy.

Table 35 Information elements in Storage-report structure

7.1.3.6 Usage Tracking Report

The usage tracking report transaction (see Figure 28: Usage Tracking Report transaction) is used by the DCD Client to report usage statistics to the DCD Server and is triggered when one of the custom conditions (e.g. low memory) occurred on the client device, upon predefined schedule, or if requested by the DCD Server. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

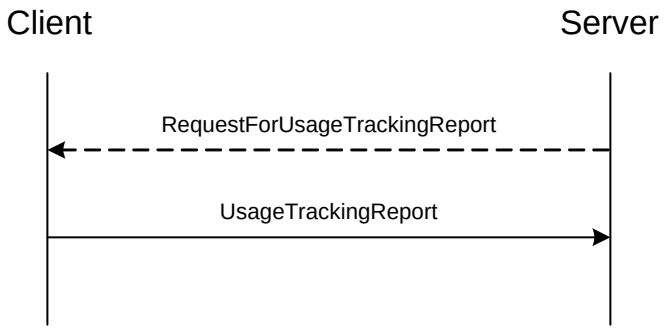


Figure 28: Usage Tracking Report transaction

7.1.3.6.1 Message and Information Elements

Message	Implementation	Direction
RequestForUsageTrackingReport	Optional	Client ← Server
UsageTrackingReport	Mandatory	Client → Server

Table 36 Message directions for Usage Tracking Report transaction

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the

			current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Application-ID	Optional	String	Application identifier of the DCD-Enabled Client Application to which this message is related. If absent, this message is used to request reports for all registered applications.
Usage-report-server-address	Optional	String	The address (URL) of the "usage tracking" server assigned by the DCD Service Provider. This address is used by the DCD Client to report usage tracking for channels via the DCD-3 interface.
Report-policy	Optional	List of Structures	Policy defining how statistics should be collected and reported. If absent, the DCD Client should immediately respond with a UsageTrackingReport based upon the currently active policies. If the policy is present but empty (i.e. contains only the Policy-ID), the policy should be removed.

Table 37 Information elements in RequestForUsageTrackingReport Message

Information Element	Req	Type	Description
Policy-ID	Mandatory	String	Identifier of the policy for correlation purposes, when usage tracking reports are sent. Policies are established by being included in a RequestForUsageTrackingReport. Policies may expire based upon their defined schedule, or be explicitly expired by being omitted from a later RequestForUsageTrackingReport. If the policy is present but empty (i.e. contains only the Policy-ID), the policy should be removed.
Channel-ID-Filter	Optional	String	A regular expression used as a filter to track content items by Channel ID metadata attribute contains this string. If present, tracking based upon the other policy filters is limited to content of this channel only. If not present, tracking based upon the other policy filters applies to content of all channels. If no other policy filters are indicated, tracking applies to all content items of the channel.
Content-ID-Filter	Optional	String	A regular expression used as a filter to track content items by Content ID.
Content-Type-Filter	Optional	String	A regular expression used as a filter to track content items by Content Type.
Mime–Type-Filter	Optional	String	A regular expression used as a filter to track content items by MIME type.

Information Element	Req	Type	Description
Tracking-Schedule	Optional	String	Tracking start and stop time (comma separated), expressed in UTC format. If not specified, tracking should start immediately.
Reporting-Schedule	Optional	String	Reporting start time expressed in UTC format, and report frequency in minutes. If not specified, tracking schedule is DCD Client-implementation specific.

Table 38 Information elements in Report-policy structure

Minimally supported regular expression subset SHALL include the following, which can be arranged in arbitrary sequence:

- "*" = match to any set of characters
- "?" = match to any single character
- "string" = match to a specified string

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.
Report-Data	Mandatory	List of Structures	Reported data. The structure depends on XML namespace and schema provided by client vendors.

Table 39 Information elements in UsageTrackingReport message

Information Element	Req	Type	Description
Policy-ID	Conditional	String	Identifier of the policy.
Content-Usage	Mandatory	List of Structures	List of Content IDs and timestamps for policy-related content items accessed. Note: how "access" is determined is implementation specific.

Table 40 Information elements in Report-Data structure

Information Element	Req	Type	Description
Content-ID	Optional	String	ID to uniquely identify content inside the Service Provider domain.
Channel-ID	Optional	String	ID to uniquely identify the channel for which the statistics apply
Subscription-ID	Optional	String	ID to uniquely identify the group of subscribers for which the statistics apply
Application-ID	Optional	String	ID to uniquely identify the application for which the statistics apply
Timestamp	Optional	String	The local device timestamp of when the content was

Information Element	Req	Type	Description
			first accessed. Should be present if Content ID is present.
Usage-Count	Mandatory	Integer	Number of times policy-related content items were accessed.

Table 41 Information elements in Content-Usage structure

7.1.3.7 Channel Subscription

The DCD Client subscribes to the DCD Channel using Channel Subscription transaction (see Figure 29: Channel Subscription). The transaction is triggered by the DCD Enabled Client Application subscribing to the DCD Channel over DCD-CAR interface. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

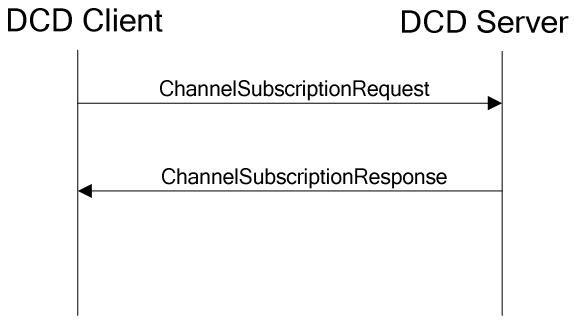


Figure 29: Channel Subscription

7.1.3.7.1 Message and Information Elements

Message	Implementation	Direction
ChannelSubscriptionRequest	Mandatory	Client → Server
ChannelSubscriptionResponse	Mandatory	Client ← Server

Table 42 Message directions in Channel Subscription transaction

Information Element	Req	Type	Description
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Channel-Metadata	Conditional	Structure	Information about the channel being subscribed to. Includes at minimum the Channel-ID. May include attributes for

			subscription personalization. See Section 13.4 for applicable subset of metadata. Present only if Content-Address is not.
Selected-Purchase-Option	Conditional	String	Purchase-Option-Id of the selected purchase option for this channel. It is present if the Purchase-Options metadata was present in the Channel Metadata
Subscription-IDs	Conditional	String	Comma-separated list of subscription identifiers for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. Present if included in the associated DCD-CAR SubscriptionRequest message.
Content-Address	Conditional	String	Address of external content (e.g. RSS URL, content provider reference to a channel etc). Present only if Channel-Metadata is not.

Table 43 Information elements in ChannelSubscriptionRequest Message

Information Element	Req	Type	Description
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Channel-Metadata	Optional	Structure	Additional information about the channel being subscribed to. For successful responses, may include client-relevant Channel Metadata parameters related to the subscribed channel. See Section 13.4 for applicable subset of metadata.

Table 44 Information elements in ChannelSubscriptionResponse Message

7.1.3.8 Channel Unsubscription

The DCD Client unsubscribes from the DCD Channel using Channel Unsubscription transaction (see Figure 30: Channel Unsubscription message flow). The transaction is triggered by the DCD Enabled Client Application unsubscribing from the DCD Channel over DCD-CAR interface. For simplification, Channel Unsubscription transaction is assumed synchronous in DCD version 1.0. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

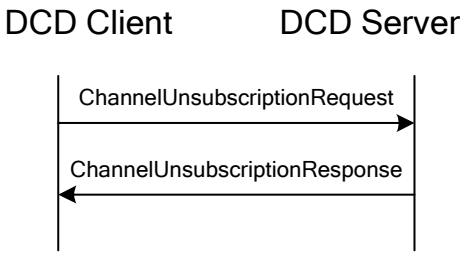


Figure 30: Channel Unsubscription message flow

7.1.3.8.1 Message and Information Elements

Message	Implementation	Direction
ChannelUnsubscriptionRequest	Mandatory	Client → Server
ChannelUnsubscriptionResponse	Mandatory	Client ← Server

Table 45 Message directions in Channel Unsubscription Message

Information Element	Req	Type	Description
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the DCD Server Domain.
Channel-ID	Mandatory	String	ID to uniquely identify a channel inside the DCD Server Domain.

Table 46 Information elements in ChannelUnsubscriptionRequest message

Information Element	Req	Type	Description
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the DCD Server Domain.

Table 47 Information elements in ChannelUnsubscriptionResponse message

7.1.3.8.2 Channel Unsubscription by DCD Server

The Channel Unsubscription Notification transaction (see Figure 31: Channel Unsubscription Notification message flow) is used by the DCD Server to notify unsubscription from the DCD Channel to which the DCD Client is subscribed.

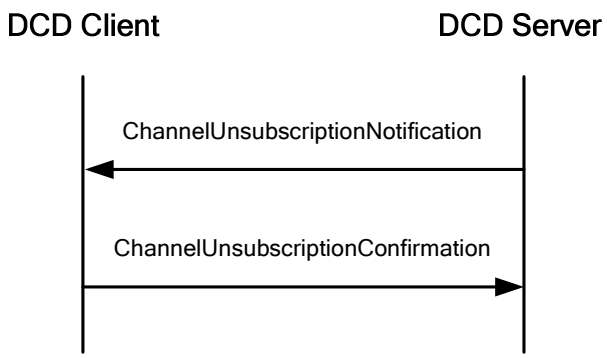


Figure 31: Channel Unsubscription Notification message flow

Message	Implementation	Direction
ChannelUnsubscriptionNotification	Mandatory	Server → Client
ChannelUnsubscriptionConfirmation	Mandatory	Server ← Client

Table 48 Message directions for Channel Unsubscription Notification

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier of the current session. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier augmented by the index of message within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-ID	Mandatory	String	Channel identifier as provided by DCD Server
Reason	Optional	String	Free form text describing the reason

Table 49 Information elements in ChannelUnsubscriptionNotification Message

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier of the current session. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier augmented by the index of message within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 50 Information elements in ChannelUnsubscriptionConfirmation Message

7.1.3.9 Channel Subscription Notification

The Channel Subscription Notification transaction (see Figure 32: Subscription Notification message flow) is used to validate subscription established outside the DCD enabler. The transaction is triggered by the DCD Content Provider notifying of the subscription established outside the DCD enabler (i.e. to confirm this subscription with the user). The confirmation message is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

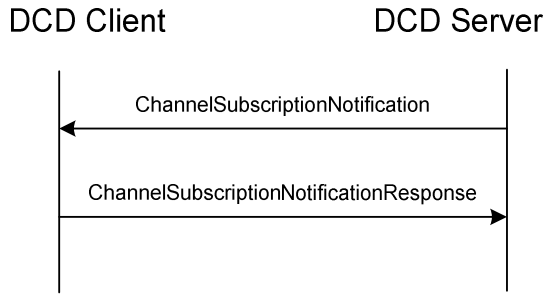


Figure 32: Subscription Notification message flow

7.1.3.9.1 Message and Information Elements

Message	Implementation	Direction
ChannelSubscriptionNotification	Mandatory	Server → Client
ChannelSubscriptionNotificationResponse	Mandatory	Server ← Client

Table 51 Message directions in channel subscription notification messaging

Information Element	Req	Type	Description
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the DCD Server Domain. This attribute is present for messages sent over point-to-point bearers.
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related. Identifies target application for the subscribed channel.
Subscription-ID	Conditional	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. SHALL be present if issued by the Content Provider during subscription established outside the DCD enabler.
Channel-Metadata	Mandatory	Structure	Channel metadata for newly subscribed channel (see Section 8.2). See Section 13.4 for applicable subset of metadata.

Table 52 Information elements in ChannelSubscriptionNotification message

Information Element	Req	Type	Description
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the DCD Server Domain.
Selected-Purchase-Option	Conditional	String	Purchase-Option-Id of the selected purchase option for this channel. It is present if the Purchase-Options metadata was present in the channel metadata sent in the SubscriptionNotification message
Channel-Metadata	Conditional	Structure	Channel Metadata parameters relevant to the DCD Server. It is present if personalization parameters have been specified by the DCD Enabled Client Application. See Section 13.4 for applicable subset of metadata.
Subscription-Declined	Conditional	Boolean	Conditional and mutually exclusive with "Selected-Purchase-Option" and "Channel-Metadata"

Table 53 Information elements in ChannelSubscriptionNotificationResponse message

7.1.3.10 Channel Discovery

Channel Discovery transactions are used to communicate channel availability across the DCD enabler. The initial set of channels matching Application Profile preferences (e.g. application ID, content type, etc.) can be provided by the DCD Server in application registration response (see Section 7.1.3.3). Due to the dynamic nature of the DCD enabler, the new channels get registered with and unused or outdated channels get deregistered from the DCD Server. Channel discovery transactions communicate these changes to the DCD Client and to the DCD Enabled Client Application (via channel discovery transactions over DCD-CADE interface) and could be initiated by the DCD Server or the DCD Client.

7.1.3.10.1 Channel Discovery Push

The Channel Discovery Push transaction (see Figure 33: Channel Discovery Push) is used by the DCD Server to notify the DCD Client about the change in channel offering and to deliver these changes via push or broadcast bearers.

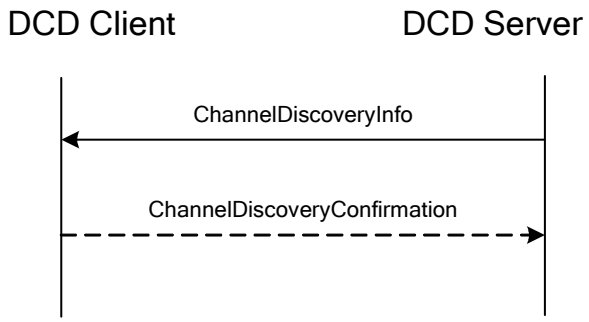


Figure 33: Channel Discovery Push

7.1.3.10.2 Message and Information Elements

Message	Implementation	Direction
ChannelDiscoveryInfo	Mandatory	Client ← Server
ChannelDiscoveryConfirmation	Optional	Client → Server

Table 54 Message directions for Channel Discovery Push

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channels-added	Conditional	List	List of Channel Metadata (see Section 8.2) for newly registered channels. At least “channel-id” and “channel-name” attributes are required to be present. May contain additional metadata attributes as per Section 13.4. Conditional on new channel additions since last update
Channels-removed	Conditional	List	List of channel IDs for deregistered channels Conditional on channel deregistrations since last update
Channels-updated	Conditional	List	List of Channel Metadata updates for existing channels. At least “channel-id” and one other attribute that has been updated are required to be present. May contain additional metadata attributes as per Section 13.4. Conditional on channel changes since last update.

Table 55 Information elements in ChannelDiscoveryInfo message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 56 Information elements in ChannelDiscoveryConfirmation message

7.1.3.10.3 Channel Discovery Notification and Pull

The Channel Discovery Notification transaction (see Figure 34: Channel Discovery Notification / Pull Transactions) is used by the DCD Server to notify the DCD Client about the change in channel offering. Upon notification via push or broadcast bearers, the DCD Server expects the DCD Client to retrieve the changes in the consequent request. Alternatively, the DCD Client may retrieve channel updates without prior notification (e.g. on an arbitrary schedule, upon application request, etc.). The Channel Discovery Notification and Pull transactions are not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

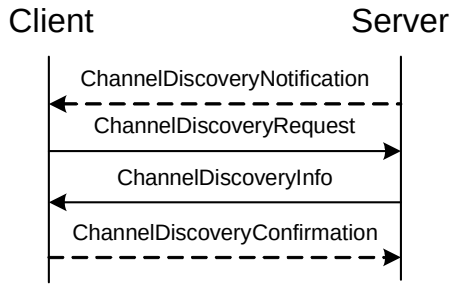


Figure 34: Channel Discovery Notification / Pull Transactions

7.1.3.10.4 Message and Information Elements

Message	Implementation	Direction
ChannelDiscoveryNotification	Optional	Client ← Server
ChannelDiscoveryRequest	Mandatory	Client → Server
ChannelDiscoveryInfo (see Section 7.1.3.10.1)	Mandatory	Client ← Server
ChannelDiscoveryConfirmation (see Section 7.1.3.10.1)	Optional	Client → Server

Table 57 Message directions for Channel Discovery Notification and Pull

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction

			identifier offset is unique within the current session and identical for all messages within the transaction.

Table 58 Information elements in ChannelDiscoveryNotification message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 59 Information elements in ChannelDiscoveryRequest message

7.1.3.11 Channel Suspend and Resume

7.1.3.11.1 Channel Suspend Request

The Channel Suspend Request transaction (see Figure 35: Channel Suspend Request Transactions) is used by the DCD Client to request the DCD Server to suspend content delivery over one or all DCD Channels. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

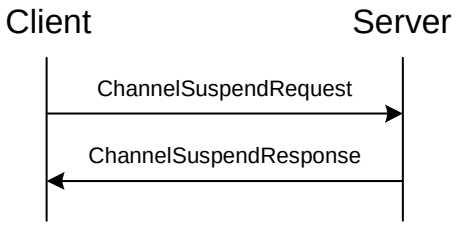


Figure 35: Channel Suspend Request Transactions

7.1.3.11.2 Message and Information Elements

Message	Implementation	Direction
ChannelSuspendRequest	Mandatory	Client → Server
ChannelSuspendResponse	Mandatory	Client ← Server

Table 60 Message directions for Channel Suspend Request

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and

			identical for all messages within the transaction.
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels are affected by this transaction.

Table 61 Information elements in ChannelSuspendRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 62 Information elements in ChannelSuspendResponse message

7.1.3.11.3 Channel Resume Request

The Channel Resume Request transaction (see Figure 36: Channel Resume Request Transactions) is used by the DCD Client to request the DCD Server resume content delivery over one or all suspended DCD Channels. This transaction is not applicable if the uplink from the DCD Client to the DCD Server is unavailable (e.g. in receive-only terminal).

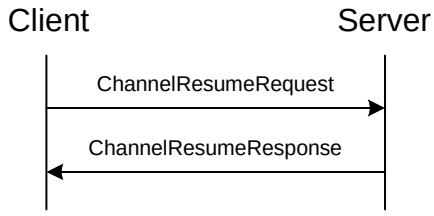


Figure 36: Channel Resume Request Transactions

7.1.3.11.4 Message and Information Elements

Message	Implementation	Direction
ChannelResumeRequest	Mandatory	Client → Server
ChannelResumeResponse	Mandatory	Client ← Server

Table 63 Message directions for Channel Resume Request

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and

			identical for all messages within the transaction.
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels are affected by this transaction.

Table 64 Information elements in ChannelResumeRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 65 Information elements in ChannelResumeResponse message

7.1.3.11.5 Channel Suspend Notification

The Channel Suspend Notification transaction (see Figure 37: Channel Suspend Notification Transactions) is used by the DCD Server to notify the DCD Client about suspension of content delivery over one or all DCD Channels.

Figure 37: Channel Suspend Notification Transactions

7.1.3.11.6 Message and Information Elements

Message	Implementation	Direction
ChannelSuspendNotification	Mandatory	Client ← Server

Table 66 Message directions for Channel Suspend Notification

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name.

			This parameter is present when CBS bearer is used for this transaction.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels are affected by this transaction.

Table 67 Information elements in ChannelSuspendNotification message

7.1.3.11.7 Channel Resume Notification

The Channel Resume Notification transaction (see Figure 38: Channel Resume Notification Transactions) is used by the DCD Server to notify the DCD Client about resumption of content delivery over one or all suspended DCD Channels.

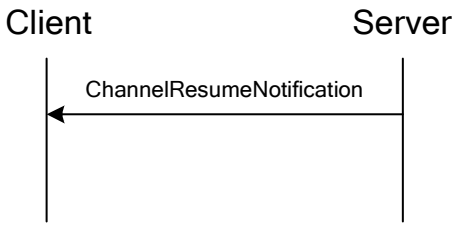


Figure 38: Channel Resume Notification Transactions

7.1.3.11.8 Message and Information Elements

Message	Implementation	Direction
ChannelResumeNotification	Mandatory	Client ← Server

Table 68 Message directions for Channel Resume Notification

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and

			identical for all messages within the transaction.
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels are affected by this transaction.

Table 69 Information elements in ChannelResumeNotification message

7.1.3.12 Content Repair

DCD channel content delivery can be interrupted during the lifespan of a channel. This may occur due to various reasons such as interference in the wireless space, loss of coverage etc. As the DCD Server cannot keep track of the states of all DCD Clients, when a DCD channel content delivery is interrupted the DCD Client requests a Content Repair request to the DCD Server. The Content Repair transaction is described in Figure 39: Content Repair Transactions. The Content Repair operation consists of the following messages:

- ContentRepairRequest from the DCD Client to the DCD Server;
- ContentRepairResponse from the DCD Server to the DCD Client.
- ContentDeliveryConfirmation from the DCD Client to the DCD Server (see Section 7.1.1.1 for details)

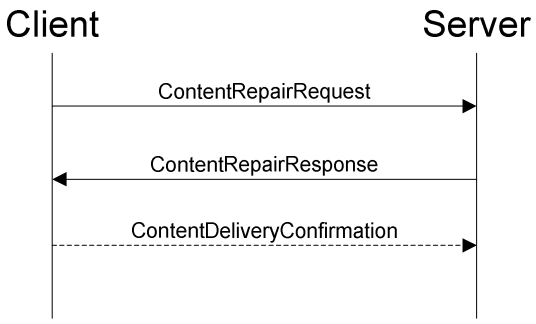


Figure 39: Content Repair Transactions

The DCD Client sends a ContentRepairRequest to the DCD Sever on the following events:

- When a content item is only partially received or;
- When a delivery is interrupted.

7.1.3.12.1 Message and Information Elements

Message	Implementation	Direction
ContentRepairRequest	Mandatory	DCD Client → DCD Server
ContentRepairResponse	Mandatory	DCD Client ←DCD Server
ContentDeliveryConfirmation (see Section 7.1.1.1)	Mandatory	DCD Client → DCD Server

Table 70 Message directions for Content Repair

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain. If there is an established Session, this attribute is present.
Message-ID	Mandatory	String	Message Identifier. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the

			current session and identical for all messages within the transaction.
Failed-Message-ID	Conditional	String	Message Identifier of failed message. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction. If the Failed-Message-ID is set, the Failed-Content-ID should be empty.
Failed-Content-ID	Conditional	String	Content Identifier of failed content. If the Failed-Content-ID is set, the Failed-Message-ID should be empty.
Channel-ID	Conditional	String	Channel Identifier. If the Failed-Content-ID is set, the Channel-ID should be set.
Bytes-Received	Mandatory	Integer	Number of bytes successfully received.

Table 71 Information elements in ContentRepairRequest message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier. The Session-ID is unique within the service provider domain. If there is an established Session, this attribute is present.
Message-ID	Mandatory	String	Message Identifier. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Content-Package	Mandatory	Data Structure	Requested content package for repair. This will either be the actual content or the reference to the link to the content.

Table 72 Information elements in ContentRepairResponse message

7.1.3.13 Channel Metadata Update

The DCD Server may remove or update Channel Metadata fields. This transaction could be triggered by Client Activation or at any time within the Session lifespan by the DCD Server as result of Service enhancement, optimization or maintenance activities.

When a Channel Metadata update is received by the DCD Client it should perform differential update of relevant fields with the provided data. If Channel Metadata field is not included in metadata update received by the DCD Client it should be interpreted as no change. If a metadata update field is empty it should be interpreted as the removal of that particular field. Otherwise, the DCD Client should update associated Channel Metadata field with the value provided in metadata update.

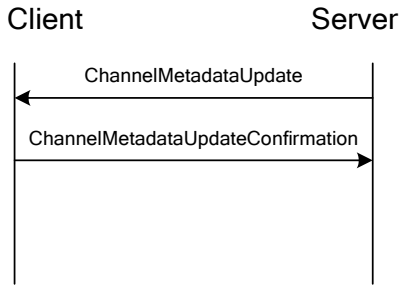


Figure 40: Channel Metadata Update

7.1.3.13.1 Message and Information Elements

Message	Implementation	Direction
ChannelMetadataUpdate	Mandatory	Client ← Server
ChannelMetadataUpdateConfirmation	Mandatory	Client → Server

Table 73 Message directions for Channel Provisioning from the DCD Server

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier of the current session. The Session-ID is unique within the service provider domain. This attribute is present for messages sent over point-to-point bearers.
Broadcast-Service-ID	Conditional	String	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the current transaction. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name. This parameter is present when CBS bearer is used for this transaction.
Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
Channel-Metadata	Mandatory	Structure	The subset of Channel Metadata that needs to be updated. See Section 13.4 for applicable subset of metadata.

Table 74 Information elements in ChannelMetadataUpdate message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session. The Session-ID is unique within the service provider domain.

Message-ID	Mandatory	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 75 Information elements in ChannelMetadataUpdateConfirmation message

7.1.3.14 Connection Profile Update

The DCD Server may add, remove or update DCD-3 Connection Profile. This transaction could be triggered by Client Activation, Application Registration procedure or at any time within the Session lifespan by the DCD Server as result of Service enhancement, optimization or maintenance activities.

When a Connection Profile Update is received by the DCD Client it should perform differential update of relevant fields with the provided data. If Connection Profile sub element or attribute are not included in Connection Profile Update received by the DCD Client it should be interpreted as no change. If a Connection Profile sub element or attribute are empty it should be interpreted as the removal of that particular element or attribute, if DCD-3 connection profile name is present in the message and the DCD-3 connection profile element is not present in the message it should be interpreted as removal of this specific DCD-3 connection profile Otherwise, the DCD Client should update associated Connection Profile sub element or attribute with the value provided in Connection Profile Update.

Default DCD-3 Connection profile can be updated using this message by keeping the DCD-3 connection profile name element in the message empty.

Through this mechanism the DCD Server can update preloaded DCD-3 Connection Profile parameters even in case that Application Registration did not happen yet.

After this transaction, the DCD Server can deactivate the previous session connected with the DCD Client and activate the new session using the updated Connection Profile, or the DCD Client can do that depending on the Service Policy.

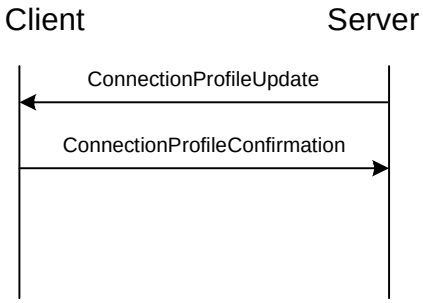


Figure 41: Connection Profile Update

7.1.3.14.1 Message and Information Elements

Message	Requirement	Direction
ConnectionProfileUpdate	Mandatory	Client ← Server
ConnectionProfileConfirmation	Mandatory	Client → Server

Table 76 Message directions for Connection Profile Provisioning from the DCD Server

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Message Identifier. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.
dcd-3-connection-profile-name	Conditional	String	Name of a locally configured DCD-3 connection profile. If not present, the message applies to the default dcd-3-connection-profile
dcd-3-connection-profile	Conditional	Structure	Specific DCD-3 connection profile to use. Omitted from the transaction if this connection profile should be removed.

Table 77 Information elements in ConnectionProfileUpdate message

Information Element	Req	Type	Description
Session-ID	Mandatory	String	Session identifier of the current session. The Session-ID is unique within the service provider domain.
Message-ID	Mandatory	String	Message Identifier. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction.

Table 78 Information elements in ConnectionProfileConfirmation message

7.1.4 Interface Extension

In order to differentiate services from different vendors and to achieve backward compatibility, an extended set of functions and parameters could be used by vendors. The definition of such extensions is out of scope of DCD 1.0.

Both DCD Client and DCD Server SHALL ignore unrecognized extensions without generating an error.

7.2 Interfaces Exposed to Content Providers

While association of the DCD Service Provider and the DCD Content Provider is a business process that is out of scope, the technical realization of this association (e.g. registration, content publication, etc.) is in scope for DCD. The current specification only deals with the requirements imposed on the DCD Server and the DCD Client, and it limits the scope of the external interfaces to the transactions that are essential for internal operations of the enabler. Due to the nature of the external interfaces, this DCD specification can only define a list of transactions and specify data parameters exchanged over these transactions. Implementation requirements stated in each of the following subsections only apply to the DCD Server, unless stated otherwise.

7.2.1 DCD-CPR

7.2.1.1 Channel Subscription with the DCD Content Provider

The DCD Server notifies a DCD Content Provider about subscription to a channel provided by the DCD Content Provider. This notification allows the DCD Content Provider to monitor channel use and validate subscribers, if required. The need for DCD Content Provider notification upon channel subscription is indicated in the channel metadata attributes provided at DCD Channel registration. For simplicity, the interaction between the DCD Server and DCD Content Provider is assumed synchronous.

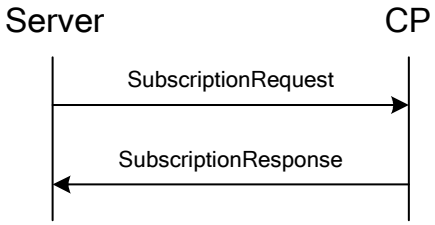


Figure 42: Channel Subscription with the DCD Content Provider

7.2.1.1.1 Message and Information Elements

Message	Implementation	Direction
SubscriptionRequest	Mandatory	Content Provider ← DCD Server
SubscriptionResponse	Mandatory	Content Provider → DCD Server

Table 76 Messages between the DCD Content Provider and the DCD Server

Information Element	Req	Type	Description
Channel-Metadata	Mandatory	Structure	Information about the channel being subscribed to. Includes at minimum the Channel-ID. May include attributes for subscription personalization. See Section 13.4 for applicable subset of metadata.
Subscriber-Info	Mandatory	String	Information about subscribing user / device. This parameter is formatted according to the XML schema published by the Service Provider.
Selected-Purchase-Option-ID	Conditional	String	Purchase-Option-Id of the selected purchase option for this channel. It is present if purchase options were defined in the Channel Metadata by the Content Provider and if the Content Provider handles charging.
Subscription-ID	Conditional	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. Conditional upon being included in the subscription request over DCD-3 interface.

Table 77 Information elements in SubscriptionRequest message

Information Element	Req	Type	Description
Channel-Metadata	Optional	Structure	Information about the channel being subscribed to. Includes at minimum the Channel-ID. For successful responses, may include attributes for subscription personalization. See Section 13.4 for applicable subset

			of metadata. When this attribute conflicts with the values provided by the DCD Enabled Client Application, the application provided values take priority.

Table 78 Information elements in SubscriptionResponse message

7.2.1.2 Channel Unsubscription notification to the DCD Content Provider

The DCD Server notifies a DCD Content Provider about unsubscription from a channel provided by the DCD Content Provider. This notification allows the DCD Content Provider to monitor channel use, if required. The notification upon channel unsubscription is issued if the need for subscription notification is specified in the Channel Metadata.

Figure 43: Channel unsubscription notification to the DCD Content Provider

7.2.1.2.1 Message and Information Elements

Message	Implementation	Direction
Unsubscription Notification	Mandatory	Content Provider ← DCD Server

Table 79 Message from the DCD Server to the DCD Content Provider

Information Element	Req	Type	Description
Subscriber-Info	Mandatory	String	Information about unsubscribing user / device. This parameter is formatted according to the XML schema published by the Service Provider.
Channel-ID	Mandatory	String	Channel identifier for subscribed channel

Table 80 Information elements in UnsubscriptionNotification message

7.2.1.3 Channel Subscription Notification by the DCD Content Provider

Channel Subscription Notification transaction (see Figure 44: Channel subscription notification by the DCD Content Provider) is used by the DCD Content Provider to notify the DCD Server about the DCD Channel subscription established outside the DCD Domain. This notification allows the DCD enabler to validate subscription with the DCD Enabled Client Application

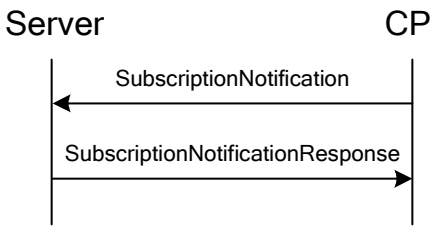


Figure 44: Channel subscription notification by the DCD Content Provider

7.2.1.3.1 Message and Information Elements

Message	Implementation	Direction
SubscriptionNotification	Mandatory	Content Provider → DCD Server
SubscriptionNotificationResponse	Mandatory	Content Provider ← DCD Server

Table 81 Messages between the DCD Content Provider and the DCD Server

Information Element	Req	Type	Description
Delivery-Endpoint-Info	Mandatory	String	Information about delivery endpoint (e.g. device and / or user identity, target application, etc.) provided by the subscriber during subscription established outside the DCD enabler. This parameter is formatted according to the XML schema published by the Service Provider.
Channel-Metadata	Mandatory	Structure	Information about the channel being subscribed to. Includes at minimum the Channel-ID. May include attributes for subscription personalization. See Section 13.4 for applicable subset of metadata.
Subscription-ID	Conditional	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. Conditional upon being issued by the DCD Content Provider during the subscription established outside the DCD enabler.

Table 82 Information elements in SubscriptionNotification message

Information Element	Req	Type	Description
Channel-Metadata	Conditional	Structure	Channel Metadata parameters relevant to the DCD Content Provider. It is present if personalization parameters have been specified by the DCD Enabled Client Application upon subscription validation. See Section 13.4 for applicable subset of metadata.
Selected-Purchase-Option-ID	Conditional	String	Purchase-Option-Id of the selected purchase option for this channel. It is present if the Purchase-Options metadata was present in the channel metadata sent in the DCD-CPR SubscriptionNotification message and if the Content Provider handles charging
Subscription-Declined	Conditional	Boolean	Conditional and mutually exclusive with "Selected-Purchase-Option" and "Channel-Metadata"

Table 83 Information elements in SubscriptionNotificationResponse message

7.2.1.4 Update Subscription notification by the DCD Content Provider

Update Subscription Notification transaction (see Figure 45: Update subscription notification by the DCD Content Provider) is used by the DCD Content Provider to assign or unassign subscription identifier to the particular user (i.e. add or remove user to the subscriber group) in order to facilitate subscription personalization. This transaction enables DCD Server to group channel subscribers according to the subscription preferences provided to DCD Content Provider by the DCD Enabled Client Application. It allows the DCD Content Provider to publish the content by the subscription identifier.

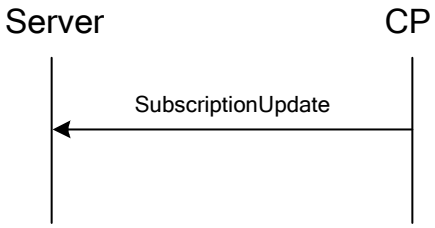


Figure 45: Update subscription notification by the DCD Content Provider

7.2.1.4.1 Message and Information Elements

Message	Implementation	Direction
SubscriptionUpdate	Mandatory	Content Provider → DCD Server

Table 84 Message from the DCD Content Provider to the DCD Server

Information Element	Req	Type	Description
Subscriber-Info	Mandatory	String	Information about subscribing user/device. This parameter is formatted according to the XML schema published by the Service Provider.
Channel-ID	Mandatory	String	Channel identifier for the subscribed channel
Subscription-ID	Mandatory	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider.
Action	Mandatory	String	Indicates whether the subscriber should be added or removed from the group (i.e. “add” or “remove”)

Table 85 Information elements in SubscriptionUpdate message

7.2.1.5 Channel Registration

The Channel Registration transaction (see Figure 46: Channel Registration Transaction) is used by the DCD Content Provider to register new channels with the DCD enabler and to supply Channel Metadata for these channels. Optionally, the DCD Server may provide the DCD Content Provider with channel reference obtained from the DCD Content Provider during channel discovery outside the DCD enabler and to prompt this DCD Content Provider to register referenced channel with the DCD enabler.

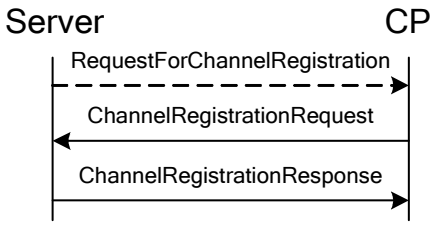


Figure 46: Channel Registration Transaction

7.2.1.5.1 Message and Information Elements

Message	Implementation	Direction
RequestForChannelRegistration	Optional	Content Provider ←DCD Server
ChannelRegistrationRequest	Mandatory	Content Provider → DCD Server
ChannelRegistrationResponse	Mandatory	Content Provider ←DCD Server

Table 86 Message directions for channel registration between Content Provider and DCD Server

Information Element	Req	Type	Description
Channel-Reference	Mandatory	String	Arbitrary string with the reference to Content Provider's channel. Supplied by the Content Provider during channel discovery outside the DCD enabler.

Table 87 Information elements in RequestForChannelRegistration

Information Element	Req	Type	Description
Channel-Metadata	Mandatory	Structure	Channel metadata (see Section 8.2). See Section 13.4 for applicable subset of metadata.

Table 88 Information elements in ChannelRegistrationRequest message

Information Element	Req	Type	Description
Channel-ID	Mandatory	String	ID of the registered channel as assigned by the DCD Server
Push-Publication-Address	Optional	URI	The address (URI) where the Content Provider can push channel updates.

Table 89 Information elements in ChannelRegistrationResponse message

7.2.1.6 Channel Deregistration

The Channel Deregistration transactions (see Figure 47: Channel Deregistration by the DCD Server and Figure 48: Channel Deregistration by the DCD Content Provider) could be initiated by the DCD Server or the DCD Content Provider.

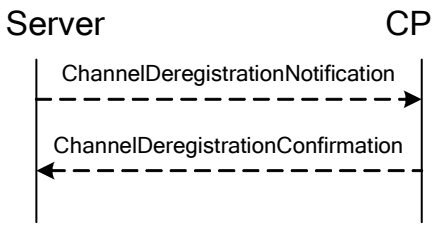


Figure 47: Channel Deregistration by the DCD Server

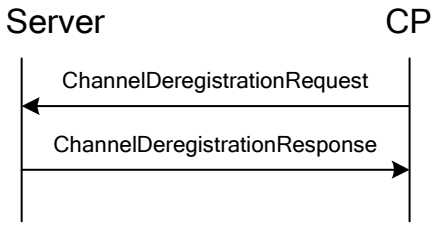


Figure 48: Channel Deregistration by the DCD Content Provider

7.2.1.6.1 Message and Information Elements

Message	Implementation	Direction
ChannelDeregistrationNotification	Optional	Content Provider ←DCD Server
ChannelDeregistrationConfirmation	Optional	Content Provider → DCD Server
ChannelDeregistrationRequest	Mandatory	Content Provider → DCD Server
ChannelDeregistrationResponse	Mandatory	Content Provider ←DCD Server

Table 90 Message Directions for Channel Deregistration between Content Provider and DCD Server

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of Channel identifier as provided by DCD Server
Reason	Optional	String	Free formed text describing the reason

Table 91 Information Elements in ChannelDeregistrationNotification Message

Information Element	Req	Type	Description

Table 92 Information Elements in ChannelDeregistrationConfirmation Message

Information Element	Req	Type	Description
Channel-ID	Mandatory	String	Channel identifier as provided by DCD Server
Reason	Optional	String	Free formed text describing the reason

Table 93 Information Elements in ChannelDeregistrationRequest Message

Information Element	Req	Type	Description

Table 94 Information Elements in ChannelDeregistrationResponse Message

7.2.1.7 Usage Report

The usage report transaction (see Figure 49: Usage Report transaction) is used by the DCD Server to report usage statistics to the DCD Content Provider and may be triggered by the DCD Content Provider.

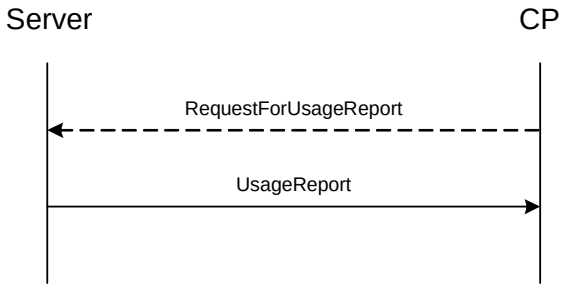


Figure 49: Usage Report transaction

7.2.1.7.1 Message and Information Elements

Message	Implementation	Direction
RequestForUsageReport	Mandatory	Server ← Content Provider
UsageReport	Mandatory	Server → Content Provider

Table 95 Message directions for Usage Report transaction

Information Element	Req	Type	Description
Channel-IDs	Optional	String	Comma separated list of Channel identifiers as provided by DCD Server. If not present all channels are targeted
Content-IDs	Optional	String	Comma separated list of Content-IDs of the specific content items requested. If not present all contents are targeted
Subscriber-Infos	Optional	List of Strings	List of Subscriber Info strings. This SHALL NOT be present if "Subcription-IDs" element is present
Subscription-IDs	Optional	String	Comma separated list of Subscription identifiers. This SHALL NOT be present if "Subscriber-Infos" element is present
Application-IDs	Optional	String	Comma separated list of application identifiers. If absent, all applications are targeted.

Table 96 Information elements in RequestForUsageReport Message

Information Element	Req	Type	Description
Report-Data	Mandatory	List of Structures	Reported data as described in 7.1.3.6.1

Table 97 Information elements in UsageReport message

7.2.2 DCD-CPDE

7.2.2.1 Channel Suspend and Resume

7.2.2.1.1 Channel Suspend Request

The Channel Suspend Request transaction (see Figure 50: Channel Suspend Request transaction) is used by the DCD Server to request the DCD Content Provider suspend content delivery over one or all DCD Channels offered by the DCD Content Provider.

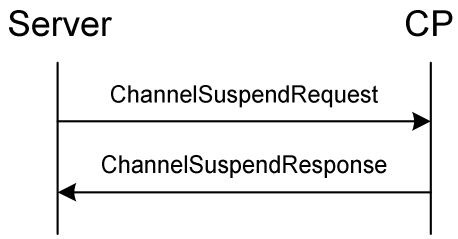


Figure 50: Channel Suspend Request transaction

7.2.2.1.2 Message and Information Elements

Message	Implementation	Direction
ChannelSuspendRequest	Mandatory	Server → Content Provider
ChannelSuspendResponse	Mandatory	Server ← Content Provider

Table 98 Message directions for Channel Suspend Request

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels from this content provider are affected by this transaction.

Table 99 Information elements in ChannelSuspendRequest message

Information Element	Req	Type	Description

Table 100 Information elements in ChannelSuspendResponse message

7.2.2.1.3 Channel Resume Request

The Channel Resume Request transaction (see Figure 51: Channel Resume Request transaction) is used by the DCD Server to request the DCD Content Provider resume content delivery over one or all DCD Channels offered by the DCD Content Provider.

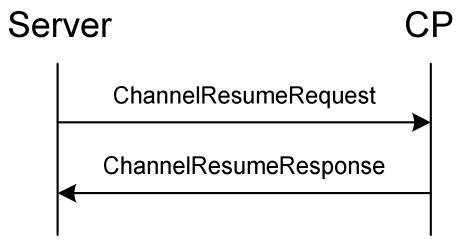


Figure 51: Channel Resume Request transaction

7.2.2.1.4 Message and Information Elements

Message	Implementation	Direction
ChannelResumeRequest	Mandatory	Server → Content Provider
ChannelResumeResponse	Mandatory	Server ← Content Provider

Table 101 Message directions for Channel Resume Request

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels offered by the content provider are affected by this transaction.

Table 102 Information elements in ChannelResumeRequest message

Information Element	Req	Type	Description

Table 103 Information elements in ChannelResumeResponse message

7.2.2.1.5 Channel Suspend Notification

The Channel Suspend Notification transaction (see Figure 52: Channel Suspend Notification transaction) is used by the DCD Content Provider to notify the DCD Server about suspension of content delivery over one or all DCD Channels.

Figure 52: Channel Suspend Notification transaction

7.2.2.1.6 Message and Information Elements

Message	Implementation	Direction
ChannelSuspendNotification	Mandatory	Server ← Content Provider

Table 104 Message directions for Channel Suspend Notification

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels offered by the content provider are affected by this transaction.

Table 105 Information elements in ChannelSuspendNotification message

7.2.2.1.7 Channel Resume Notification

The Channel Resume Request transaction (see Figure 53: Channel Resume Notification transaction) is used by the DCD Content Provider to notify the DCD Server about resumption of content delivery over one or all DCD Channels. As opposed to Channel Suspend Notification, this transaction may require a confirmation message to indicate readiness to obtain DCD Content.

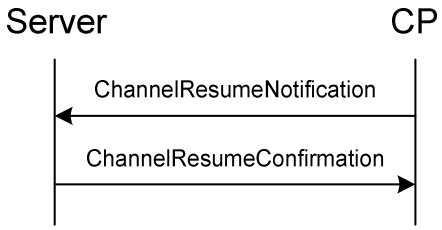


Figure 53: Channel Resume Notification transaction

7.2.2.1.8 Message and Information Elements

Message	Implementation	Direction
ChannelResumeNotification	Mandatory	Server ← Content Provider
ChannelResumeConfirmation	Mandatory	Server → Content Provider

Table 106 Message directions for Channel Resume Request transaction

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels of this content provider are affected by this transaction.

Table 107 Information elements in ChannelResumeNotification message

Information Element	Req	Type	Description

Table 108 Information elements in ChannelResumeConfirmation message

7.2.2.2 Content Update

The Content Update transaction (see Figure 54: Content Update transaction) allows DCD Server to request content from the DCD Content Provider or to submit content from the DCD-Enabled Client Application.

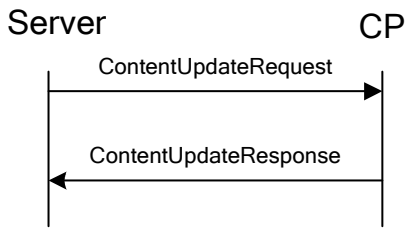


Figure 54: Content Update transaction

7.2.2.2.1 Message and Information Elements

Message	Implementation	Direction
ContentUpdateRequest	Mandatory	Server → Content Provider
ContentUpdateResponse	Mandatory	Server ← Content Provider

Table 109 Message Directions for Content Update transaction

Information Element	Req	Type	Description
Channel-ID	Mandatory	String	Channel identifier for the channel offered by the content provider
Subscription-IDs	Conditional	String	Comma separated list of subscription identifiers. Present, if requesting user has subscription IDs associated with this channel.
Submit-Package	Conditional	Opaque Data	Content to be submitted. SHALL be present if the ContentUpdateRequest was initiated by a DCD-1 ContentSubmitRequest. User identity and other applicable metadata (as created by submitting application), if needed, can be included by the application in the Submit Package.

Table 110 Information elements in ContentUpdateRequest message

Information Element	Req	Type	Description
Content-Package	Conditional	List of Data Structures	One or more content packages. The content package consists of content payload and content metadata. Content payload could be empty if content-address parameter of the content metadata contains the address where the content could be retrieved (e.g. URI). This parameter is present when the DCD Content Provider has updated content to return to the DCD enabler.

Table 111 Information elements in ContentUpdateResponse message

7.2.2.3 Content Publication

The Content Publication (see Figure 55: Content Publication) allows the DCD Content Provider to publish DCD Content at the DCD Server.

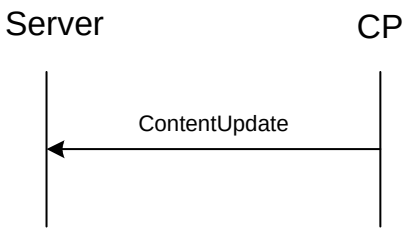


Figure 55: Content Publication

7.2.2.3.1 Message and Information Elements

Message	Implementation	Direction
ContentUpdate	Mandatory	Server ← Content Provider

Table 112 Message directions for Content Publication

Information Element	Req	Type	Description
Content-Package	Mandatory	List of Data Structures	One or more content packages. The content package consists of content payload and content metadata. Content payload could be empty if content-address parameter of the content metadata contains the address where the content could be retrieved (e.g. URI). If Channel-ID element is present in the message all items in Content-Package element belong to the same channel (identified by Channel-ID element).
Channel-ID	Optional	String	ID of the channel related to the content package. If absent, the content metadata in the Content-Package element SHALL contain Channel-ID attribute.
Subscription-IDs	Conditional	String	Comma separated list of subscription identifiers. Present, if content publication is targeted to the particular groups of subscribers.

Table 113 Information elements in ContentUpdate Message

7.3 Interfaces Exposed to DCD-Enabled Client Application

The implementation of interactions between the DCD Enabled Client Applications and the DCD Client depend on the device execution environment, deployment policy, and other factors that are not controlled by the DCD enabler. The deployment models (e.g. DCD Client embedded in the DCD Enabled Client Application) and implementation specifics (e.g. programming languages, operating systems) could differ from device to device, therefore the device side external interfaces are only expressed through a set of supported transactions and data parameters exchanged over these transactions. Implementation requirements stated in each of the following subsections only apply to the DCD Client, unless stated otherwise. Note that while these interface functions are listed in the DCD Client SCR table and associated with Section 6 functional requirements, the expected deployment and implementation variations effectively prevent interoperability testing of the interfaces.

7.3.1 DCD-CAR

For simplification and due to the nature of the mobile device execution environment, all transactions over DCD-CAR interface are considered to be synchronous.

7.3.1.1 Registration of DCD-Enabled Client Application

The DCD-Enabled Client Application Registration transaction (see Figure 56: DCD-Enabled Client Application Registration with DCD Client) allows the DCD-Enabled Client Application to submit its Application Profile (see Section 8.1) to the DCD enabler.

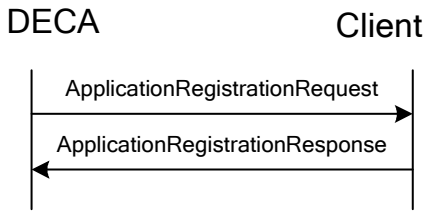


Figure 56: DCD-Enabled Client Application Registration with DCD Client

7.3.1.1.1 Message and Information Elements

Message	Implementation	Direction
ApplicationRegistrationRequest	Mandatory	DCD Client ←DCD-Enabled Client Application
ApplicationRegistrationResponse	Mandatory	DCD Client → DCD-Enabled Client Application

Table 114 Message directions for DCD-Enabled Client Application Registration

Information Element	Req	Type	Description
Application-Profile	Mandatory	Structure	Contains application preferences. The details of Application-Profile are defined in Section 8.1.

Table 115 Information elements in ApplicationRegistrationRequest message

Information Element	Req	Type	Description
Channel-discovery-information	Conditional	String	Location of channel discovery information provided upon successful registration. This is present if the channel metadata was provided to the DCD Client as part of the registration response over DCD-3 interface.

Table 116 Information elements in ApplicationRegistrationResponse message

7.3.1.2 Deregistration of DCD-Enabled Client Application

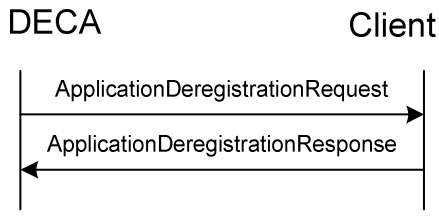


Figure 57: DCD-Enabled Client Application Deregistration with DCD Client

7.3.1.2.1 Message and Information Elements

Message	Implementation	Direction
ApplicationDeregistrationRequest	Mandatory	DCD Enabled Client Application → DCD Client
ApplicationDeregistrationResponse	Mandatory	DCD Enabled Client Application ← DCD Client

Table 117 Message directions for DCD Enabled Client Application Deregistration

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.

Table 118 Information elements in ApplicationDeregistrationRequest message

Information Element	Req	Type	Description

Table 119 Information elements in ApplicationDeregistrationResponse message

7.3.1.3 Channel Subscription

The Channel Subscription transaction (see Figure 58: Subscription to the DCD Channel) used by the DCD-Enabled Client Application to subscribe to DCD Channels offered by the DCD enabler.

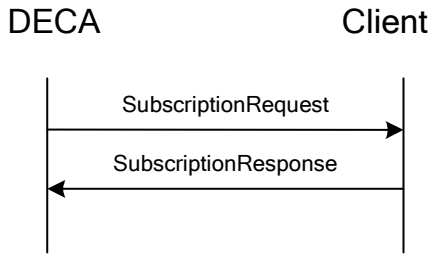


Figure 58: Subscription to the DCD Channel

7.3.1.3.1 Message and Information Elements

Message	Implementation	Direction
SubscriptionRequest	Mandatory	DCD-Enabled Client Application → DCD Client
SubscriptionResponse	Mandatory	DCD-Enabled Client Application ← DCD Client

Table 120 Message directions in Subscription messaging

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.
Channel-Metadata	Conditional	Structure	Channel metadata associated with the subscribed channel. If present in the message, contains at least channel-id attribute. See Section 13.4 for applicable subset of metadata. Present only if Content Address is not.
Selected-Purchase-Option	Conditional	String	Purchase-Option-Id of the selected purchase option for this channel. It is present if the Purchase-Options metadata was present in the

			channel metadata
Content-Address	Conditional	String	Content address (e.g. RSS URL), content provider reference to a channel etc. Present only if Channel-Metadata is not.
Subscription-ID	Conditional	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. Conditional upon being issued by the DCD Content Provider during the subscription established outside the DCD enabler, e.g. through the browser.

Table 121 Information elements in SubscriptionRequest message

Information Element	Req	Type	Description
Channel-Metadata	Conditional	Structure	Channel metadata associated with the subscribed channel. Present, if the subscription is successful and the information was not already available to the application through the channel guide provided upon registration or channel discovery. See Section 13.4 for applicable subset of metadata.

Table 122 Information elements in SubscriptionResponse message

7.3.1.4 Channel Subscription Update

The Channel Subscription Update transaction (see Figure 59: Subscription Update transaction) used by the DCD Enabled Client Application to update its subscription to the DCD Channels by updating channel metadata associated with the subscribed channel. The Channel Subscription Update transaction could be triggered by Channel Metadata update transaction if the modified channel metadata affects the original subscription.

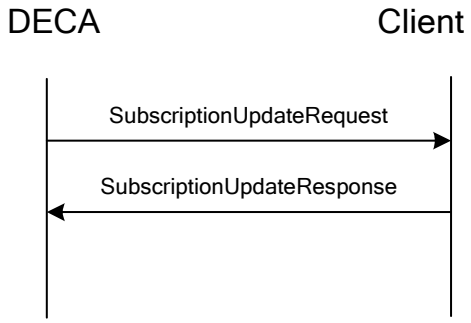


Figure 59: Subscription Update transaction

7.3.1.4.1 Message and Information Elements

Message	Implementation	Direction
SubscriptionUpdateRequest	Mandatory	DCD-Enabled Client Application → DCD Client
SubscriptionUpdateResponse	Mandatory	DCD-Enabled Client Application ← DCD Client

Table 123 Message directions in Channel Subscription Update Transaction

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.
Channel-Metadata	Mandatory	Structure	Channel Metadata representing updated subscription/personalization options for the DCD enabler. See Section 13.4 for applicable subset of metadata.
Subscription-ID	Conditional	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. Provided, if issued in the original subscription request.

Table 124 Information elements in SubscriptionUpdateRequest message

Information Element	Req	Type	Description

Table 125 Information elements in SubscriptionUpdateResponse message

7.3.1.5 Channel Unsubscription

The Channel Unsubscription transaction (see Figure 60: Channel Unsubscription message flow) allows DCD Enabled Client Application to unsubscribe from the subscribed DCD Channel.

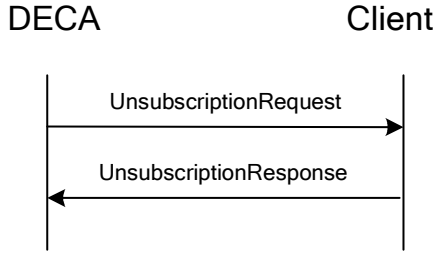


Figure 60: Channel Unsubscription message flow

7.3.1.5.1 Message and Information Elements

Message	Implementation	Direction
UnsubscriptionRequest	Mandatory	DCD Enabled Client Application → DCD Client
UnsubscriptionResponse	Mandatory	DCD Enabled Client Application ← DCD Client

Table 126 Message directions in Channel Unsubscription messaging

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.
Channel-ID	Mandatory	String	Channel Identifier for the unsubscribed channel

Table 127 Information elements in UnsubscriptionRequest message

Information Element	Req	Type	Description

Channel-ID	Mandatory	String	Channel identifier for subscribed channel

Table 128 Information elements in UnsubscriptionResponse message

7.3.1.6 Channel Unsubscription Notification

The Channel Unsubscription transaction (see Figure 61: Unsubscription Notification message flow) is used by the DCD Client when the DCD Channel subscribed by the DCD Enabled Client Application has been deregistered, or other reasons on the server side. The transaction is triggered when DCD Client receives Unsubscription Notification (see Section 7.1.3.8) from the DCD Server.

Figure 61: Unsubscription Notification message flow

7.3.1.6.1 Message and Information Elements

Message	Implementation	Direction
UnsubscriptionNotification	Mandatory	DCD Client → DCD Enabled Client Application

Table 129 Message directions for Unsubscription Notification

Information Element	Req	Type	Description
Channel-ID	Mandatory	String	Channel Identifier for the unsubscribed channel
Reason	Optional	String	Free form text describing the reason

Table 130 Information elements in UnsubscriptionNotification message

7.3.1.7 Subscription Validation

The Subscription Validation transaction (see Figure 62: Subscription Validation transaction) is used to validate with the DCD Enabled Client Application a subscription established outside the DCD enabler. The transaction is triggered by the DCD Client when received Channel Subscription Notification (see Section 7.1.3.9) from the DCD Server. By responding with SubscriptionValidationResponse message, the DCD Enabled Client Application accepts or rejects the subscription.

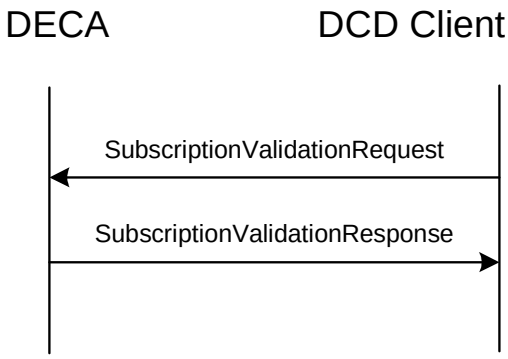


Figure 62: Subscription Validation transaction

7.3.1.7.1 Message and Information Elements

Message	Implementation	Direction
SubscriptionValidationRequest	Mandatory	DCD Client → DCD Enabled Client Application
SubscriptionValidationResponse	Mandatory	DCD Client ← DCD Enabled Client Application

Table 131 Message directions in Subscription Validation messaging

Information Element	Req	Type	Description
Subscription-ID	Conditional	String	Subscription identifier for grouping of channel subscribers according to subscription preferences managed by the DCD Content Provider. Conditional upon presence in SubscriptionNotification message over DCD-3 interface.
Channel-Metadata	Mandatory	Structure	Metadata of the subscribed channel. Includes at minimum the Channel-ID. See Section 13.4 for applicable subset of metadata.

Table 132 Information elements in SubscriptionValidationRequest message

Information Element	Req	Type	Description
Channel-Metadata	Mandatory	Structure	Metadata of the subscribed channel. Includes at minimum the Channel-ID. May include attributes for subscription personalization. See Section 13.4 for applicable subset of metadata.
Selected-Purchase-Option	Conditional	String	Purchase-Option-Id of the selected purchase option for this channel. It is present if the Purchase-Options metadata was present in the channel metadata sent in the SubscriptionValidationRequest message
Subscription-Declined	Conditional	Boolean	Conditional and mutually exclusive with “Selected-Purchase-Option” and “Channel-Metadata”.

Table 133 Information elements in SubscriptionValidationResponse message

7.3.1.8 Channel Metadata Update

The ChannelMetadataUpdate message is used to notify the DCD Enabled Client Application about updates to Channel Metadata for subscribed channels.

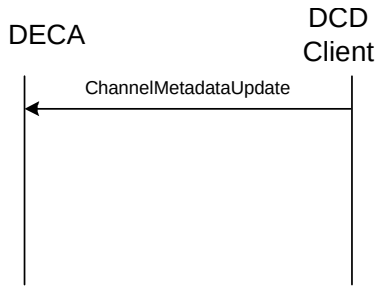


Figure 63: Channel Metadata Update

7.3.1.8.1 Message and Information Elements

Message	Implementation	Direction
ChannelMetadataUpdate	Mandatory	DCD Enabled Client Application ← DCD Client

Table 134 Message directions for Channel Metadata Update messaging

Information Element	Req	Type	Description
Channel-Metadata	Mandatory	Structure	The subset of Channel Metadata that needs to be updated. See Section 13.4 for applicable subset of metadata.

Table 135 Information elements in ChannelMetadataUpdate message

7.3.2 DCD-CADE

7.3.2.1 Channel Discovery

Channel discovery transaction (see Figure 64: Channel Discovery transaction) allows DCD Enabled Client Application to retrieve "channel guide" from the DCD Client.

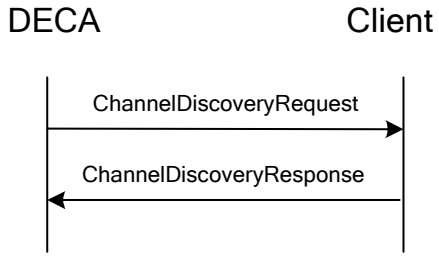


Figure 64: Channel Discovery transaction

7.3.2.1.1 Message and Information Elements

Message	Implementation	Direction

ChannelDiscoveryRequest	Mandatory	DCD Enabled Client Application → DCD Client
ChannelDiscoveryResponse	Mandatory	DCD Enabled Client Application ← DCD Client

Table 136 Message directions for Channel Discovery Transaction

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.

Table 137 Information elements in ChannelDiscoveryRequest message

Information Element	Req	Type	Description
Channel-guide	Mandatory	List	The list of channel metadata structures for the channels offered to the application. The channel metadata in the channel guide contains the subset of channel metadata parameters applicable to the DCD Enabled Client Application (see Sections 8.2 and 13.4 for details)

Table 138 Information elements in ChannelDiscoveryResponse message

7.3.2.2 Content Delivery

Content Delivery transaction (see Figure 65: Content Delivery with prior Content Request and Figure 66: Content Notification or Content Delivery without prior Content Request) allows the DCD-Enabled Client Application to request the DCD Content from the DCD Client. The DCD Content could be already stored at the DCD Client (e.g. delivered over DCD-2 interface). The DCD-Enabled Client Application may request content by channel identifier, content identifier, or content address. If none of these is specified in the Content Request message, the DCD Client returns all content available for the requesting application.

Alternatively, the DCD Client may send the content to the DCD-Enabled Client Application without prior Content Request.

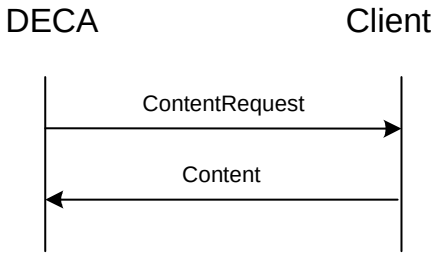


Figure 65: Content Delivery with prior Content Request

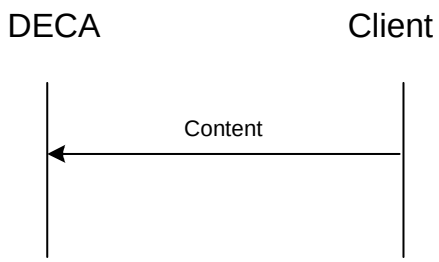


Figure 66: Content Notification or Content Delivery without prior Content Request

7.3.2.2.1 Message and Information Elements

Message	Implementation	Direction
ContentRequest	Mandatory	DCD Client ← DCD Enabled Client Application
Content	Mandatory	DCD Client → DCD Enabled Client Application

Table 139 Message directions for Content Delivery transaction

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.
Channel-IDs	Optional	String	Comma separated list of channel identifiers for which currently available content is requested.
Content-id	Optional	String	content-id of the specific content item requested.
Content-address	Optional	String	content-address of the specific content item requested.

Table 140 Information elements in ContentRequest message

Information Element	Req	Type	Description
Content-Metadata	Mandatory	Structure	DCD Content Metadata (see Section 8.3.2) as provided by the DCD enabler.
Content	Optional	Opaque Data	Content for the DCD-Enabled Client Application (opaque to the DCD Enabler).

Table 141 Information elements in Content message

7.3.2.3 Content Submission

The Content Submission transaction is illustrated in Figure 67: Content Submission Transaction with DCD Content returned to the DCD Enabled Client Application and Figure 68: Content Submission Transaction without DCD Content returned to the DCD Enabled Client Application. The transaction is initiated by DCD Enabled Client Applications. DCD Enabled Client Applications can send a ContentSubmitRequest message to the DCD Client for arbitrary application-specific purposes, which are transparent to the DCD enabler.

The Content Submission transaction may or may not result in the DCD Content returned to the DCD Client in response. If there is DCD Content returned in response to content submission, the transaction consists of the following messages:

- ContentSubmitRequest from the DCD Enabled Client Application to the DCD Client
- Content (see Section 7.3.2.2) from the DCD Client to the DCD Enabled Client Application

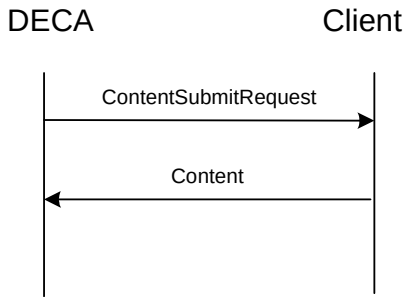


Figure 67: Content Submission Transaction with DCD Content returned to the DCD Enabled Client Application

If there is no DCD Content returned in response to content submission, the transaction consists of the following messages:

- ContentSubmitRequest from the DCD Enabled Client Application to the DCD Client
- ContentSubmitConfirmation from the DCD Client to the DCD Enabled Client Application

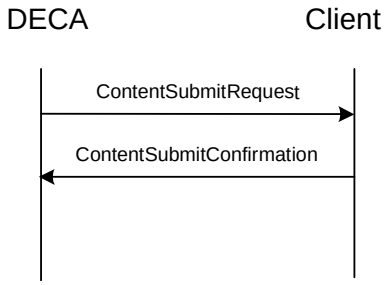


Figure 68: Content Submission Transaction without DCD Content returned to the DCD Enabled Client Application

7.3.2.3.1 Message and Information Elements

Message	Implementation	Direction
ContentSubmitRequest	Mandatory	DCD Client ← DCD Enabled Client Application
ContentSubmitConfirmation	Mandatory	DCD Client → DCD Enabled Client Application
Content "(see 7.3.2.2.1)"	Mandatory	DCD Client → DCD Enabled Client Application

Table 142 Message directions for Content Submission transaction

Information Element	Req	Type	Description
Application-ID	Mandatory	String	Application identifier of the DCD-Enabled Client Application to which this message is related.
Channel-ID	Mandatory	String	ID of channel related to the submitted content.
Submit-Address	Optional	URI	URI to which the DCD enabler should deliver the submit request.
Submit-Package	Mandatory	Opaque Data	Content to be submitted. Submit-Package can contain user identity and/or other metadata that is targeted for the Content Provider and opaque for the DCD Enabler.

Table 143 Information elements in ContentSubmitRequest message

Information Element	Req	Type	Description

Table 138 Information elements in ContentSubmitConfirmation message

7.3.2.4 Channel Suspend and Resume

7.3.2.4.1 Channel Suspend Request

The Channel Suspend Request transaction (see Figure 69: Channel Suspend Request transaction) is used by the DCD Enabled Client Application to request content delivery suspension for one or all DCD Channels consumed by this application

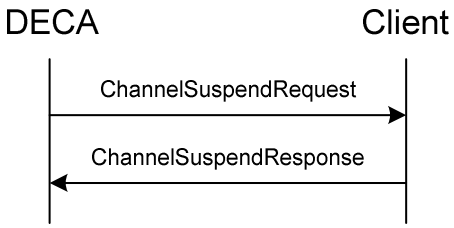


Figure 69: Channel Suspend Request transaction

7.3.2.4.2 Message and Information Elements

Message	Implementation	Direction
ChannelSuspendRequest	Mandatory	DCD Enabled Client Application → DCD Client
ChannelSuspendResponse	Mandatory	DCD Enabled Client Application ← DCD Client

Table 144 Message directions for Channel Suspend Request

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all non-emergency channels consumed by this application are affected by this transaction.

Table 145 Information elements in ChannelSuspendRequest message

Information Element	Req	Type	Description

Table 146 Information elements in ChannelSuspendResponse message

7.3.2.4.3 Channel Resume Request

The Channel Resume Request transaction (see Figure 70: Channel Resume Request transaction) is used by the DCD Enabled Client Application to request content delivery resumption for one or all suspended DCD Channels

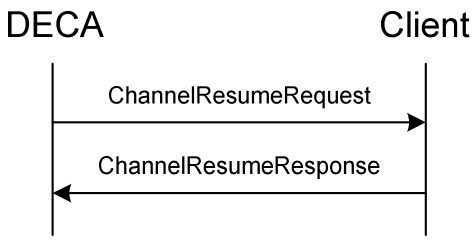


Figure 70: Channel Resume Request transaction

7.3.2.4.4 Message and Information Elements

Message	Implementation	Direction
ChannelResumeRequest	Mandatory	DCD Enabled Client Application → DCD Client
ChannelResumeResponse	Mandatory	DCD Enabled Client Application ← DCD Client

Table 147 Message directions for Channel Resume Request

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all suspended channels are affected by this transaction.

Table 148 Information elements in ChannelResumeRequest message

Information Element	Req	Type	Description

Table 149 Information elements in ChannelResumeResponse message

7.3.2.4.5 Channel Suspend Notification

The Channel Suspend Notification Transaction is used to notify the content delivery suspension for one or all suspended DCD Channels to the relevant DCD Enabled Client Application(s) when the DCD Client receives the ChannelSuspendNotification Message from the DCD Server.

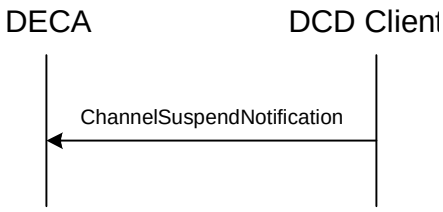


Figure 71: Channel Suspend Notification transaction

7.3.2.4.6 Message and Information Elements

Message	Implementation	Direction
ChannelSuspendNotification	Mandatory	DCD Client → DCD Enabled Client Application

Table 150 Message directions for Channel Suspend Notification

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all channels consumed by this application are affected by this transaction.

Table 151 Information elements in ChannelSuspendNotification message

7.3.2.4.7 Channel Resume Notification

The Channel Resume Notification Transaction is used to notify the content delivery resumption for one or all suspended DCD Channels to the relevant DCD Enabled Client Application(s) when the DCD Client receives the ChannelResumeNotification Message from the DCD Server.

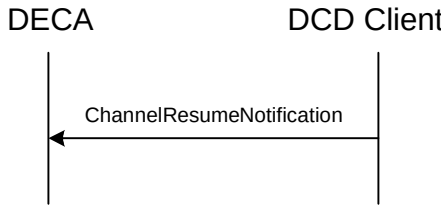


Figure 72: Channel Resume Notification transaction

7.3.2.4.8 Message and Information Elements

Message	Implementation	Direction
ChannelResumeNotification	Mandatory	DCD Client → DCD Enabled Client Application

Table 152 Message directions for Channel Resume Notification

Information Element	Req	Type	Description
Channel-IDs	Mandatory	String	Comma separated list of channel identifiers. If empty, all suspended channels are affected by this transaction.

Table 153 Information elements in ChannelResumeNotification message

7.3.2.5 Channel Discovery Info

The ChannelDiscoveryInfo message (see Figure 73: Channel Discovery Info Message) is sent from the DCD Client to the DCD Enabled Client Application to communicate availability of new channels and removal of or updates to the existing channels. ChannelDiscoveryInfo message is triggered by a channel discovery push transaction over the DCD-3 interface (see Section 7.1.3.10.1), or a channel discovery pull transaction (see Section 7.1.3.10.3).

Channel discovery push will not be available for the DCD Enabled Client Application if channel-discovery-notification attribute in the Application Profile disallows it (see Section 8.1).

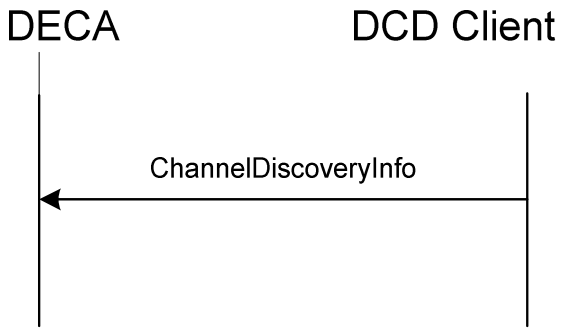


Figure 73: Channel Discovery Info Message

7.3.2.5.1 Message and Information Elements

Message	Implementation	Direction
ChannelDiscoveryInfo	Mandatory	DCD Client → DCD Enabled Client Application

Table 154 Message directions for Channel Discovery Info

Information Element	Req	Type	Description
Channels-added	Conditional	List	The list of channel metadata structures for newly registered channels. The channel metadata is limited to the subset of channel metadata parameters applicable to the DCD Enabled Client Application (see Section 8.2 for details). At least "channel-id" and "channel-name" attributes are required to be present. May contain additional metadata attributes as per Section 13.4. Conditional on new channel additions since last update
Channels-removed	Conditional	List	List of channel IDs for deregistered channels. Conditional on channel deregistrations since last update
Channels-updated	Conditional	List	The list of channel metadata structures for updated channels. The channel metadata is limited to the subset of channel metadata parameters applicable to the DCD Enabled Client Application (see Section 8.2 for details). At least "channel-id" and one other attribute that has been updated are required to be present. May contain additional metadata attributes as per Section 13.4. Conditional on channel changes since last update.

Table 155 Information elements in ChannelDiscoveryInfo message

8. DCD Metadata

8.1 Application Profile

8.1.1 Application Profile

This section describes the Application Profile purpose, attributes, and relevance to DCD entities and interfaces. Other sections describe the requirements specific to DCD operations (e.g. registration) that include exchange and use of Application Profiles and their attributes.

The Application Profile is defined in [DCD-AD] as: *"the set of static definitions and rules that allow the DCD enabler to handle the delivery of the DCD Content for a particular DCD-Enabled Client Application. The AP consists of the collection of Channel Metadata plus application specific parameters common for multiple DCD Channels (e.g. application-id). The AP facilitates the processing of dynamic metadata (i.e. Content Metadata)."*

Application Profiles are provided by DCD-Enabled Client Applications during the registration process, so the DCD Client and DCD Server can be aware of the specific needs of the DCD-Enabled Client Applications. Application Profiles can be updated by DCD-Enabled Client Applications as their needs change, through a process of re-registration.

Following is an informative description of Application Profile attributes:

- **Application ID**: an identifier for the application. This may be defined with strict semantics or more loosely, e.g. as a conventional user-agent header. DCD Clients and DCD Servers can use this string for compatibility checks or service-matching, as described in Section 5.2
- **Application description:** the "friendly" name of the application.
- **DCD-3 connection profile name:** a pre-configured DCD-3 connection profile to use for the DCD-3 interface. DCD Clients may be configured with a set of DCD-3 connection profiles that can allow DCD-Enabled Client Applications to select the connection profile to use via this attribute. This attribute is mutually exclusive with the specific set of DCD-3 connection profile attributes. If neither this attribute nor a specific set of DCD-3 connection profile attributes is specified, the DCD Client will use a default DCD-3 connection profile.
- **DCD-3 connection profile:** a set of parameters that define how the DCD Client interacts with the DCD Server for the DCD-3 interface. When DCD Enabled Client Application submits the DCD-3 connection profile, the DCD Client validates submitted connection profile and responds with activation and session establishment with the DCD Server specified by this profile. This behaviour is subject to device/client vendor and service provider policies. The DCD-3 connection profile consists of the following parameters:
    - **DCD Server address**: address (URL) of the DCD server with which the DCD Client should activate DCD. This attribute can be used when a DCD-Enabled Client Application requires service via a specific DCD Server.
    - **Proxy**: address (IP address or hostname) of the WAP proxy that should be used for transactions via the DCD-3 interface.
    - **Data connection details**: Additional bearer-network-specific connection details for the DCD-3 interface, e.g. APN, data connection username/password, etc.
    - **Broadcast profile**: Broadcast bearer connection details, e.g. cell broadcast message-identifier.
- **Channel discovery notification**: indication whether the DCD-Enabled Client Application wants explicit notification from the DCD Client for availability of new channels which are compatible with the DCD-Enabled Client Application per its Application Profile. This applies to DCD Server-initiated Content Discovery only.
- **Channel selection metadata:** a subset of Channel Metadata (see Section 8.2) that allows DCD Server and/or DCD Client to facilitate channel selection for the DCD Enabled Client Application.

8.1.2 Application Profile Elements, Attributes and Values

This section defines the Application Profile in detail.

DCD Servers and DCD Clients SHALL support DCD Application Profiles defined in accordance with the "DCD Application Profile" XML schema.

Unless otherwise specified, all Application Profile elements and attributes are mandatory for support by DCD Servers and DCD Clients, and optional for use.

Explanation of the Type in Table 156 Application Profile:

- E = Elements
- A = Attributes
- 1,2,3, etc = nesting level of the element

Application Profile basic structure:

application-profile – E
- application-id – A
- application-description – A
- channel-discovery-notification – A
- dcd-3-connection-profile-name – A
- dcd-3-connection-profile – E1
  - dcd-server-address – A
  - proxy – A
  - data-connection-details – E2
    - apn – A
    - auth-method – A
    - auth-username – A
    - auth-password – A
  - broadcast-profile – E2
    - cell-broadcast-message-id – A
    - broadcast-service-id - A
    - response-spread – A
    - bcast-access-info – E3
      - service-fragment-reference – E4
      - access-fragment – E4
      - sdp-description– E4
- dcd-channel-selection-metadata – E1

Name	Type	Cardinality	Description	Data Type	Use in DCD-3	Use in DCD-CAR
application-profile	E	1	Contains the following **attributes**: application-id application-description channel-discovery-notification dcd-3-connection-profile-name Contains the following **sub-elements**: dcd-3-connection-profile dcd-channel-selection-metadata	Structure	Mandatory	Mandatory
application-id	A	1	ID of the Application	String	Mandatory	Mandatory
application-	A	0..1	"Friendly name" of the application. May be used by	String	Not	Optional

Name	Type	Cardinality	Description	Data Type	Use in DCD-3	Use in DCD-CAR
description			the DCD Client for unspecified purposes, e.g. logging.		Allowed	
channel-discovery-notification	A	0..1	An optional attribute indicating whether the DCD-Enabled Client Application wants explicit notification from the DCD Client when new channels are available. DCD Clients can validate that notifications are sent only for channels which are compatible with the DCD-Enabled Client Application per its Application Profile. Values: 0 – False **1 – True (*)**	Boolean	Not Allowed	Optional
dcd-3-connection-profile-name	A	0..1	Name of a locally configured DCD-3 connection profile to be used, if matched by the DCD Client.	String	Not Allowed	Conditional (note 2)
dcd-3-connection-profile	E1	0..1	Specific DCD-3 connection profile to use. Contains the following **attributes**: dcd-server-address proxy Contains the following **sub-elements**: data-connection-details broadcast-profile	Structure	Not Allowed	Conditional (note 2)
dcd-server-address	A	0..1	Address (URL) of the DCD server with which the DCD Client should activate DCD.	URI	Not Allowed	Optional
proxy	A	0..1	Address (IP address or hostname) of the WAP proxy that should be used for transactions via the DCD-3 interface.	String	Not Allowed	Optional
data-connection-details	E2	0..1	Additional bearer-network-specific connection details for the DCD-3 interface, e.g. APN, data connection username / password, etc. Contains the following **attributes**: apn auth-method auth-username auth-password	Structure	Not Allowed	Optional
apn	A	0..1	Access Point Name used to establish a data connection for the interface.	String	Not Allowed	Optional
auth-method	A	0..1	Authentication method for the interface. Possible values: “none”, “digest-user”, “digest-gba”, “x509”. If this attribute is not present, method “none” is assumed.	String	Not Allowed	Optional
auth-username	A	0..1	Username for use with “digest-user” authentication method. If auth-method “digest-user” is specified and this attribute is not present, the username is to be determined through other unspecified means, e.g. user prompts.	String	Not Allowed	Optional
auth-password	A	0..1	Password for use with “digest-user” authentication method. If auth-method “digest-user” is specified and this attribute is not present, the password is to be determined through other unspecified means, e.g. user prompts.	String	Not Allowed	Optional
broadcast-profile	E2	0..1	Broadcast bearer connection details, e.g. cell broadcast message-identifier and/or BCAST access parameters. Contains the following **attributes**:	Structure	Not Allowed	Optional

Name	Type	Cardinality	Description	Data Type	Use in DCD-3	Use in DCD-CAR
			cell-broadcast-message-id response-spread broadcast-service-id Contains the following **sub-elements**: bcast-access-info			
cell-broadcast-message-id	A	0..1	The Cell Broadcast Service Message Identifier (logical Cell Broadcast Service channel) from which the DCD Client should expect DCD-3 interface data delivered via Cell Broadcast Service.	Integer	Not Allowed	Optional
response-spread	A	0..1	Period in seconds over which DCD Clients should randomize responses to DCD-2 notifications, in order to spread response load at the DCD Server.	Integer	Not allowed	Optional (when broadcast is used)
broadcast-service-id	A	0..1	A comma separated list of Broadcast-Service-IDs. Includes all Broadcast Service IDs applicable for the Application. It is present only if CBS technology is supported. Broadcast Service IDs are intended to be globally unique, and start with a registered Internet domain name.	String	Not allowed	Optional (when broadcast is used)
bcast-access-info	E3	0..1	OMA BCAST specific connection details (e.g. multicast IP, port, and TSI) for file delivery session over which the DCD Client should expect DCD-3 interface data to be delivered via OMA BCAST. Note: One of 'service-fragment-reference' or 'access-fragment' or 'sdp-description', but not more than one, SHALL be instantiated. Implementation in XML Schema should use <choice>. Contains the following **sub-elements**: service-fragment-reference access-fragment sdp-description	Structure	Not Allowed	Optional
service-fragment-reference	E4	1	URI matching the "id" attribute of the OMA BCAST Service Guide "Service" fragment for the BCAST service associated with the DCD-3 interface (see Section 5.1.2.1 of [BCAST-TS-Service_ Guide])	URI	Not Allowed	Optional
access-fragment	E4	1..n	Complete OMA BCAST Service Guide "Access" fragment as described in Section 5.1.2.4 of [BCAST-TS-Service_ Guide] containing access information for the file delivery session associated with the DCD-3 interface. Note: Applications SHOULD provide at most one access fragment per underlying BDS.	Structure	Not Allowed	Optional
sdp-description	E4	1..n	A SDP describing the connection parameters	String	Not Allowed	Optional
dcd-channel-selection-metadata	E1	0..N	A subset of Channel Metadata (see Section 8.2) that allows DCD Server and/or DCD Client to facilitate channel selection for the DCD Enabled Client Application	Structure	Conditional (note 1)	Optional

Table 156 Application Profile

Table notes:

(1) If provided in registration via DCD-CAR, has to be forwarded via DCD-3.

(2) These attributes are mutually exclusive.

(*) Default value to use when the attribute is omitted.

8.2 Channel Metadata

8.2.1 Channel Metadata

This section describes Channel Metadata purpose, attributes, and relevance to DCD entities and interfaces. Other sections describe the requirements specific to DCD operations that include exchange and use of Channel Metadata.

Channel Metadata is defined in [DCD-AD] as: *"a set of static settings and rules for handling delivery of the DCD Content for a particular channel (e.g. delivery, storage, notification rules). Channel Metadata is associated with the channel's content types."*

Content Providers are a key source of Channel Metadata, since it describes the channels that carry their services and helps DCD enabler to establish content delivery for these channels.

DCD Enabled Client Applications can submit Channel Metadata to the DCD enabler:

- As part of Registration [see Section 5.2].
- As part of Subscription [see Section 5.5], for initial subscription to a channel or to change the settings for an existing channel.

At registration, the DCD Enabled Client Applications can submit Channel Metadata parameters that facilitate channel selection (see Section 8.2.2.1.1) and at subscription, the applications can submit Channel Metadata parameters that describe delivery preferences and channel personalization (see Section 8.2.2.2.3 and 8.2.2.1.2)

Channel Metadata is grouped into logical subsets (e.g. subscription related, delivery related, etc.) based on the source of the metadata attribute and its use in different DCD transactions. Following is an informative description of Channel Metadata attributes:

- **Channel ID**: an identity, assigned by the DCD Service Provider, to uniquely identify the channel within the DCD Service Provider domain. For content delivered via the DCD-2 interface, this attribute identifies the content's relationship to a specific channel.
- **DCD interface**: a list of DCD interfaces ("DCD-1" and/or "DCD-2") used for content delivery. If this attribute is not present, both DCD-1 and DCD-2 are assumed applicable.
- **MIME types**: a list of MIME types for the content to be delivered in the channel. DCD Enabled Client Applications can include this attribute to specify what MIME types they support, or to identify the MIME types that they want/expect for the channels of interest. Content Providers can include this attribute to specify MIME types for the content in the offered channels.
- **Content types**: a list of strings that describe the type of channel content e.g. by "category", "tag" or "relation". This is a comma-separated list of tokens of undefined grammar. DCD-Enabled Client Applications can include this attribute to indicate application preferences for available channels using free-text content tags or to identify the content subset that they want/expect for selected channels during subscription personalization. Content Providers can include this attribute in Channel Metadata to describe offered channels.
- **Content availability notification**: an indication that the DCD-Enabled Client Application should be given explicit notification from the DCD Client when new content has been received for this channel, e.g. if the DCD-Enabled Client Application does not periodically check a pre-determined content storage location for new content.
- **Content storage location**: a specific location where the DCD-Enabled Client Application wants delivered content for this channel to be stored. When this attribute is provided, the storage is "co-managed" by the DCD Client and DCD-Enabled Client Application (both can update the content as needed).
- **DCD-provided storage size**: a reserved amount of content storage space (in bytes) for this channel. This attribute applies only when DCD-provided storage is used. This is mutually exclusive with content-storage-location.

- **Charging rules**: indications provided by the Content Provider regarding the charging of this channel, e.g. how the channel should be charged (based on subscription or on consumption), the entity responsible for charging. The format of this parameter can be provided by the service provider.
- **Purchase options**: list of prices and price conditions (e.g. duration or amount of data) that the DCD-Enabled Client Application should be aware of before subscribing to a channel.
- **DCD-1 connection profile name:** a pre-configured DCD-1 connection profile to use for the DCD-1 interface. DCD Clients that are configured with a set of DCD-1 connection profiles can allow DCD-Enabled Client Applications to select the connection profile to use via this attribute. This attribute is mutually exclusive with the specific set of DCD-3 connection profile attributes. If neither this attribute nor specific DCD-1 connection profile attributes are provided, the DCD Client will use a default DCD-1 connection profile.
- **DCD-1 connection profile:** a set of parameters that define how the DCD Client interacts with the DCD Server for the DCD-1 interface:
    - **DCD server address**: the address (URL) of the DCD-1 interface server assigned by the DCD Service Provider. This address is used by the DCD Client to retrieve content for channels via the DCD-1 interface.
    - **Proxy**: address (IP address or hostname) of the WAP proxy that should be used for transactions via the DCD-1 interface.
    - **Data connection details**: Additional bearer-network-specific connection details for the DCD-1 interface, e.g. APN, data connection username/password, etc.
- **DCD-2 connection profile:** a set of parameters that define how the DCD Client receives content via the DCD-2 interface for specific transports (note: WAP Push requires no special configuration). If these attributes are not present, the DCD Client will use a default DCD-2 connection profile.
    - **Cell Broadcast message ID:** the Cell Broadcast Service Message ID (logical Cell Broadcast Service channel) from which the DCD Client should expect DCD-2 interface data delivered via Cell Broadcast Service.
    - **BCAST access info:** the BCAST access parameters for the file delivery session over which DCD Clients should expect DCD-2 interface data delivered via BCAST.

8.2.2 Channel Metadata Elements, Attributes and Values

This section defines the Channel Metadata in detail.

DCD Servers and DCD Clients SHALL support DCD Channel Metadata defined in accordance with the “DCD Channel Metadata” XML schema.

Unless otherwise specified, all Channel Metadata elements and attributes are mandatory for support by DCD Servers and DCD Clients, and optional for use.

Channel Metadata can be partitioned into the following logical subsets based on the source of metadata and the use in DCD transaction:

- Application supplied Channel Metadata
- Content Provider supplied Channel Metadata

Application supplied Channel Metadata is further partitioned into the following subsets:

- Channel Selection Metadata
- Delivery Personalization Metadata

Content Provider supplied metadata could be partitioned into the following subsets:

- General Channel Metadata
- Charging Metadata
- Delivery Preferences Metadata
- Channel Publication Metadata

Some attributes of the Content Provider supplied metadata could be added or updated by the DCD Server as specified in the tables below.

DCD transactions may carry one or more subsets of Channel Metadata, detailed mapping of DCD messages to the appropriate subsets of channel metadata is specified in the DCD XML schema (see Section 9.4).

Explanation of the Type for metadata items:

- E = Elements
- A = Attributes
- 1,2,3, etc = nesting level of the element

Channel Metadata subsets have the following structure:

channel-selection-metadata – E1

- mime-types – A
- content-types – A

delivery-personalization-metadata – E1

- channel-id – A
- content-availability-notification – A
- dcd-provided-storage-size – A
- content-storage-location – A
- network-preferences – A
- deliver-when-roaming – A
- update-upon-expiration – A
- dcd-2-broadcast-profile-name – A
- dcd-2-broadcast-profile – E2
    - cell-broadcast-message-id – A
    - bcast-access-info – E3
        - service-fragment-reference – E4
        - access-fragment – E4
        - sdp-description – E4

general-channel-metadata – E1

- channel-id – A
- updated – A
- storage-reservation – A
- channel-name – A
- channel-description – A
- channel-icon – E1

    - mime-type - A
- adaptation-capability – E1
- genre – A
- parental-rating – A
- content-expiration – A
- content-protection – A
- matching-applications – A
- mime-types – A
- content-types – A
- subscription-required – A

charging-metadata – E1

- charging-rules – E2
- purchase-options – E2
    - purchase-option-id – A
    - cost-information – A
    - price – E3
        - amount – A
        - currency – A

delivery-preferences-metadata – E1

- channel-id – A
- network-preferences – A
- dcd-interface – A
- deliver-when-roaming – A
- suspend-allowed – A
- on-demand-pull-allowed – A
- pull-schedule – E2
    - pull-start-date – A
    - pull-start-time – A
    - pull-interval – A
    - pull-end-date – A
    - pull-end-time – A
- update-upon-expiration-allowed – A
- dcd-1-connection-profile-name – A
- dcd-1-connection-profile – E2
    - dcd-server-address – A
    - proxy – A
    - data-connection-details – E3
        - apn – A
        - auth-method – A
        - auth-username – A
        - auth-password – A
- dcd-2-broadcast-profile-name – A
- dcd-2-broadcast-profile – E2
    - cell-broadcast-message-id – A
    - bcast-access-info – E3
        - service-fragment-reference – E4
        - access-fragment – E4
        - sdp-description – E4

channel-publication- metadata – E1

- channel-availability-start – A

- channel-availability-end – A
- publication-methods – A
- pull-publication-address – A
- publication-schedule – E2
  - publication-start-date – A
  - publication-start-time – A
  - polling-interval – A
  - publication-end-date – A
  - publication-end-time – A
- push-schedule – E2
  - push-start-date – A
  - push-start-time – A
  - push-interval – A
  - push-end-date – A
  - push-end-time – A
- channel-personalization – E2
  - deliver-per-location – E3
  - deliver-per-presence – E3
  - deliver-per-xdms – E3
- channel-subscription – E2
  - cp-subscription-manager-address – A
  - notify-on-subscription-change – A
  - proxied-subscription – A

8.2.2.1 Channel Metadata supplied by a DCD Enabled Client Application

8.2.2.1.1 Channel Selection Metadata

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
channel-selection-metadata	E	1	Contains the following **attributes**: - mime-types - content-types	Structure	DS, DC
mime-types	A	0..1	A comma-separated list of MIME types that application is designed to support	String	DS, DC
content-types	A	0..1	A comma-separated list of strings that describe the type of channel content of interest e.g. by “category”, “tag” or "relation".	String	DS, DC

Table 157 Channel Selection Metadata

8.2.2.1.2 Delivery Personalization Metadata

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
delivery-personalization-metadata	E	1	Contains the following **attributes**: - channel-id - content-availability-notification - dcd-provided-storage-size - content-storage-location - network-preferences - deliver-when-roaming	Structure	DS, DC

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
			- update-upon-expiration - dcd-2-broadcast-profile-name Contains the following **elements**: - dcd-2-broadcast-profile		
channel-id	A	1	Channel identifier assigned by the DCD Service Provider, to uniquely identify the channel within the DCD Service Provider domain. It is provided to the application through channel discovery means.	String	DS, DC
content-availability-notification	A	0..1	An indication that the DCD Enabled Client Application should be given explicit notification from the DCD Client when new content has been received for this channel. Values: **0 – False (*)** 1 – True	Boolean	DC
dcd-provided-storage-size	A	0..1	A requested reserved amount of content storage space (in bytes) for this channel. This is mutually exclusive with content-storage-location.	Integer	DC
content-storage-location	A	0..1	A specific location provided by the DCD-Enabled Client Application where it wants the content delivered for this channel to be stored.	String	DC
network-preferences	A	0..1	Descending priority-ordered, comma-separated list of network/bearer types for use in content delivery, selected per arbitrary deployment-specific criteria for network selection (e.g. GPRS vs. UMTS vs. Wi-Fi) based on delivery cost, bandwidth, quality of service, etc. DCD Client and DCD Server apply these criteria for content delivery over DCD-1 and DCD-2 interfaces. One or more of “EDGE”, “UMTS”, “CDMA2000”, “GAN”, “WiMAX”, “LTE”, “802.11”, “CBS”, “BCAST”.	String	DS, DC
deliver-when-roaming	A	0..1	Indication that content delivery is allowed when roaming. Values: **0** – **False (*)** 1 – True	Boolean	DS, DC
update-upon-expiration	A	0..1	Indicates that the DCD Client should automatically send a content update request when all content for the channel has expired. This attribute is ignored if update-upon-expiration-allowed is “False”. Values: **0** – **False (*)** 1 – True	Boolean	DC
dcd-2-broadcast-profile-name	A	0..1	A pre-configured DCD-2 connection profile to use for the DCD-2 interface. This attribute is mutually exclusive with dcd-2-broadcast-profile.	String	DS, DC
dcd-2-broadcast-profile	E2	0..1	A set of parameters that define how the DCD Client receives content via the DCD-2 interface for specific transports (note: WAP Push requires no special configuration). Contains the following **attributes**: cell-broadcast-message-id Contains the following **sub-elements**: bcast-access-info	Structure	DC
cell-broadcast-message-id	A	0..1	The Cell Broadcast Service Message Identifier (logical Cell Broadcast Service channel) from which the DCD Client should expect DCD-2 interface data delivered via Cell Broadcast Service.	Integer	DS, DC
bcast-access-info	E4	0..1	OMA BCAST specific connection details (e.g. multicast IP, port, and TSI) for file delivery session over which the DCD Client should expect DCD-2	Structure	DS, DC

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
			interface data to be delivered via OMA BCAST Note: One of 'service-fragment-reference' or 'access-fragment' or 'sdp-description' but not more than one SHALL be instantiated. Implementation in XML Schema should use <choice>. Contains the following sub-elements: service-fragment-reference access-fragment Sdp-description		
service-fragment-reference	E5	1	URI matching the "id" attribute of the OMA BCAST Service Guide "Service" fragment for the BCAST service associated with this DCD channel (see Section 5.1.2.1 of [BCAST-TS-Service_ Guide])	URI	DS, DC
access-fragment	E5	1..n	Complete OMA BCAST Service Guide "Access" fragment as described in Section 5.1.2.4 of [BCAST-TS-Service_ Guide] containing access information for the file delivery session associated with this DCD channel Note: Applications SHOULD provide at most one access fragment per underlying BDS	Structure	DS, DC
sdp-description	E5	1..n	A SDP describing the connection parameters	String	DS, DC

Table 158 Delivery Personalization Metadata

8.2.2.2 Channel Metadata supplied by a DCD Content Provider

8.2.2.2.1 General Channel Metadata

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
general-channel-metadata	E1	1	Contains the following **attributes**: - channel-id - updated - storage-reservation - channel-name - channel-description - genre - parental-rating - content-expiration - content-protection - matching-applications - mime-types - content-types - subscription-required Contains the following **sub-elements**: channel-icon adaptation-capability	Structure	DS, DC	
channel-id	A	1	Channel identifier assigned by the DCD Service Provider, to uniquely identify the channel within the DCD Service Provider domain. It is inserted in general channel metadata by the DCD Server. Content Providers are expected to submit an empty string when registering a channel.	String	DS, DC	Yes

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
updated	A	1	Time when the channel metadata 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 in the absence of a numeric time zone offset.	dateTime	DS, DC	Yes
storage-reservation	A	0..1	Minimum device storage (in bytes) required for subscription to this channel. Can be used by DS and DC for channel selection and filtering.	Integer	DS, DC	Yes
channel-name	A	0..1	Name of the channel. Can be presented to the user in content discovery.	String	DC	Yes
channel-description	A	0..1	Text description of the channel. Can be presented to the user in content discovery.	String	DC	Yes
channel-icon	E2	0..1	Channel icon which can be displayed to the user. Can be presented to the user in content discovery. Contains the following **attributes**: mime-type	Image or URI	DC	Yes
mime-type	A	0..1	The MIME type for the channel icon to be delivered in the channel. Not included if the Channel-icon is a URI.	String	DC	Yes
adaptation-capability	E2	0..1	Compatibility with specific client capabilities (language choices, screen sizes etc). Can be used by DS or DC for channel/content selection and filtering. This parameter should be specified based on the format published by the Service Provider	String	DS, DC	Yes
genre	A	0..1	Genre of the channel content.	String	DC	Yes
parental-rating	A	0..1	Content rating per FCC "TV Parental Guidelines" or similar local regulatory requirements. Can be used by DS and DC for channel selection and filtering.	String	DS, DC	Yes
content-expiration	A	0..1	Default period (in seconds) during which content items are allowed to remain in storage.	Integer	DS, DC	Yes
content-protection	A	0..1	Indication of the content protection type applied to channel content. Can be used by DS and DC to apply content protection.	String	DS, DC	Yes
matching-applications	A	0..1	A comma separated list of Application IDs of the Applications matching the channel, if known to the Content Provider or DCD Service Provider.	String	DS, DC	No
mime-types	A	0..1	A comma-separated list of MIME types for content items in the offered channel	String	DS, DC	Yes
content-types	A	0..1	A comma-separated list of strings that describe the type of channel content e.g. by “category”, “tag” or "relation".	String	DS, DC	Yes
subscription-required	A	0..1	This attribute indicate whether DCD Client SHALL auto subscribe to this channel	Boolean	DC	No

Table 159 General Channel Metadata

8.2.2.2.2 Charging Metadata

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
charging-metadata	E1	1	Contains the following **sub-elements**: - charging-rules - purchase-options	Structure	DS, DC	
charging-rules	E2	0..1	Information that at least defines • who is responsible for the channel charging, • charging method (based on subscription or consumption) This parameter should be specified based on the format published by the Service Provider	String	DS	No
purchase-options	E2	0..N	A set of parameters that defines the purchase options for the channel. This element is not present if the charging is handled by the Content Provider. Contains the following **attributes**: - purchase-option-id - cost information Contains the following **sub-element**: - price	Structure	DC	Yes
purchase-option-id	A	1	Identifier of the purchase option, that will be used to identify the selected option	String	DC	Yes
cost-information	A	0..1	Information for presentation purpose that may contain the price and pricing conditions	String	DC	Yes
price	E3	0..1	Contains the following **attributes**: - amount - currency	Structure	DC	Yes
amount	A	1	Specifies the monetary value of the price for this purchase option	Float	DC	Yes
currency	A	1	Specifies the monetary currency codes defined in ISO 4217 international currency codes.	Integer	DC	Yes

Table 160 Charging Metadata

8.2.2.2.3 Delivery Preferences Metadata

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
delivery-preferences-metadata	E1	1	Contains the following **attributes**: - channel-id - network-preferences - dcd-interface - deliver-when-roaming - suspend-allowed - on-demand-pull-allowed - update-upon-expiration-allowed - dcd-1-connection-profile-name	Structure	DS, DC	

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
			- dcd-2-broadcast-profile -name Contains the following **sub-elements**: - pull-schedule - dcd-1-connection-profile - dcd-2-broadcast-profile			
channel-id	A	1	Channel identifier assigned by the DCD Service Provider, to uniquely identify the channel within the DCD Service Provider domain. It is inserted in delivery preferences metadata by the DCD Server.	String	DS, DC	No
network-preferences	A	0..1	Descending priority-ordered, comma-separated list of network/bearer types for use in content delivery, selected per arbitrary deployment-specific criteria for network selection (e.g. GPRS vs. UMTS vs. Wi-Fi) based on delivery cost, bandwidth, quality of service, etc. DCD Client and DCD Server apply these criteria for content delivery over DCD-1 and DCD-2 interfaces. One or more of "EDGE", "UMTS", "CDMA2000", "GAN", "WiMAX", "LTE", "802.11", "CBS", "BCAST".	String	DS, DC	No
dcd-interface	A	0..1	A comma-separated list of DCD interfaces to be used for content delivery. Values: DCD-1/HTTP DCD-1/HTTPS DCD-2/Point-To-Point DCD-2/Broadcast	String	DS, DC	No.
deliver-when-roaming	A	0..1	Content Provider supplied indication whether the content delivery is allowed when roaming. Values: **0 – False (*)** 1 – True	Boolean	DS, DC	No
suspend-allowed	A	0..1	Indication that DCD-Enabled Client Application is allowed to suspend delivery, either upon application request or based upon provided metadata. Values: 0 – False **1 – True (*)**	Boolean	DC	Yes
on-demand-pull-allowed	A	0..1	Indication that on-demand channel update requests by the DCD-Enabled Client Application are allowed. 0 – False **1 – True (*)**	Boolean	DC	Yes
pull-schedule	E2	0..N	Set of schedules upon which the DCD Client should request channel updates via DCD-1. Contains the following **attributes**: pull-start-date pull-start-time pull-interval pull-end-date pull-end-time	Structure	DC	No
pull-start-date	A	0..1	Date on which content pull should begin.	Date	DC	No
pull-start-time	A	0..1	Time at which interval pull should begin. If the time value is a clock time only (does not include a weekday or date indication), pull should begin daily at the given time.	Time	DC	No
pull-interval	A	0..1	Interval between content update requests by the DCD Client, in seconds.	Integer	DC	No
pull-end-date	A	0..1	Date on which content pull ends.	Date	DC	No
pull-end-time	A	0..1	Time at which content pull ends. If the time value is a clock time only (does not include a weekday or date indication), pull should end	Time	DC	No

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
			daily at the given time.			
pull-end	A	0..1	Time at which interval pull ends. If the time value is a clock time only (does not include a weekday or date indication), pull should end daily at the given time.	String	DC	No
update-upon-expiration-allowed	A	0..1	Indicates that the DCD Enabled Client Application is allowed to request that the DCD Client automatically send a content update request when all content for the channel has expired. Values: **0 – False (*)** 1 – True	Boolean	DC	Yes
dcd-1-connection-profile-name	A	0..1	A pre-configured DCD-1 connection profile to use for the DCD-1 interface. This attribute is mutually exclusive with dcd-1-connection-profile.	String	DC	No
dcd-1-connection-profile	E2	0..1	A set of parameters that define how the DCD Client interacts with the DCD Server for the DCD-1 interface. This attribute is mutually exclusive with dcd-1-connection-profile-name. Contains the following **attributes**: dcd-server-address proxy Contains the following **sub-elements**: data-connection-details	Structure	DC	No
dcd-server-address	A	1	The address (URL) of the DCD-1 interface server assigned by the DCD Service Provider. This address is used by the DCD Client to retrieve content for channels via the DCD-1 interface.	URI	DC	No
proxy	A	0..1	Address (IP address or hostname) of the WAP proxy that should be used for transactions via the DCD-1 interface.	String	DC	No
data-connection-details	E3	0..1	Additional bearer-network-specific connection details for the DCD-1 interface, e.g. APN, data connection username / password, etc. Contains the following **attributes**: apn auth-method auth-username auth-password	Structure	DC	No
apn	A	0..1	Access Point Name used to establish a data connection for the interface.	String	DC	No
auth-method	A	0..1	Authentication method for the interface. Possible values: “none”, “digest-user”, “digest-gba”, “x509”. If this attribute is not present, method “none” is assumed.	String	DC	No
auth-username	A	0..1	Username for use with “digest-user” authentication method. If auth-method “digest-user” is specified and this attribute is not present, the username is to be determined through other unspecified means, e.g. user prompts.	String	DC	No
auth-password	A	0..1	Password for use with “digest-user” authentication method. If auth-method “digest-user” is specified and this attribute is not present, the password is to be determined through other unspecified means, e.g. user prompts.	String	DC	No
dcd-2-broadcast-profile-name	A	0..1	A pre-configured DCD-2 connection profile to use for the DCD-2 interface. This attribute is mutually exclusive with dcd-2-	String	DS, DC	No

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities	Include In Channel Guide
			broadcast-profile.			
dcd-2-broadcast-profile	E2	0..1	A set of parameters that define how the DCD Client receives content via the DCD-2 interface for specific transports (note: WAP Push requires no special configuration). Contains the following **attributes**: cell-broadcast-message-id Contains the following **sub-elements**: bcast-access-info	Structure	DC	No
cell-broadcast-message-id	A	0..1	The Cell Broadcast Service Message Identifier (logical Cell Broadcast Service channel) from which the DCD Client should expect DCD-2 interface data delivered via Cell Broadcast Service.	Integer	DS, DC	No
bcast-access-info	E4	0..1	OMA BCAST specific connection details (e.g. multicast IP, port, and TSI) for file delivery session over which the DCD Client should expect DCD-2 interface data to be delivered via OMA BCAST Note: One of 'service-fragment-reference' or 'access-fragment' or 'sdp-description' but not more than one SHALL be instantiated. Implementation in XML Schema should use <choice>. Contains the following sub-elements: service-fragment-reference access-fragment Sdp-description	Structure	DS, DC	No
service-fragment-reference	E5	1	URI matching the "id" attribute of the OMA BCAST Service Guide "Service" fragment for the BCAST service associated with this DCD channel (see Section 5.1.2.1 of [BCAST-TS-Service_ Guide])	URI	DS, DC	No
access-fragment	E5	1..n	Complete OMA BCAST Service Guide "Access" fragment as described in Section 5.1.2.4 of [BCAST-TS-Service_ Guide] containing access information for the file delivery session associated with this DCD channel Note: Applications SHOULD provide at most one access fragment per underlying BDS	Structure	DS, DC	No
sdp-description	E5	1..n	A SDP describing the connection parameters	String	DS, DC	No

Table 161 Delivery Preferences Metadata

8.2.2.2.4 Channel Publication Metadata

The following table describes channel metadata related to channel content publication by the DCD Content Provider incl. subscription arrangements between the DCD Server and the DCD Content Provider, and the delivery personalization preferences on the Service Provider side.

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
channel-publication-	E1	1	Contains the following **attributes**:	Structure	DS

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
metadata			- channel-availability-start - channel-availability-end - publication-methods - pull-publication-address Contains the following **sub-elements**: - publication-schedule - channel-personalization - channel-subscription - push-schedule		
channel-availability-start	A	1	Time when the channel content publication will begin.	dateTime	DS
channel-availability-end	A	0..1	Time when the channel content publication will end.	dateTime	DS
publication-methods	A	1	The methods the CP will use to publish channel content to the DCD Server: “pull”: by the DCD Server pulling the updates “push”: by the CP pushing the updates	List of Strings	DS
pull-publication-address	A	0..1	The address (URI) where the DCD Server can pull channel updates.	URI	DS
publication-schedule	E2	0..N	Set of schedules upon which the DCD Server should request channel updates via DCD-CPDE. Contains the following **attributes**: publication-start-date publication-start-time polling-interval publication-end-date publication-end-time	Structure	DS
publication-start-date	A	0..1	Date on which content publication begins.	Date	DS
publication-start-time	A	0..1	Time at which interval publication begins. If the time value is a clock time only (does not include a weekday or date indication), publication begins daily at the given time.	Time	DS
polling-interval	A	0..1	Recommended interval in seconds at which Service Provider should poll for content updates for the channel (e.g. 10min for stock updates, 1 day for daily newspaper, etc.)	Integer	DS
publication-end-date	A	0..1	Date on which content publication ends.	Date	DS
publication-end-time	A	0..1	Time at which interval publication ends. If the time value is a clock time only (does not include a weekday or date indication), publication ends daily at the given time.	Time	DS
channel-personalization	E2	0..1	Contains the following **elements**: - deliver-per-location - deliver-per-presence - deliver-per-xdms	Structure	DS
deliver-per-location	E3	0..N	A rule for matching a location at which delivery should be allowed. The location rule should be specified based on the format published by the Service Provider	String	DS
deliver-per-presence	E3	0..N	A rule for allowing delivery based upon matching a subscribed Presence attribute. The rule should be specified based on the format published by the Service Provider	String	DS
deliver-per-xdms	E3	0..N	A rule for allowing delivery based upon matching a subscribed XDMS attribute. The rule should be specified based on the format published by the Service Provider	String	DS
channel-subscription	E2	0..1	Contains the following **attributes**: - cp-subscription-manager-address - notify-on-subscription-change - proxied-subscription	Structure	DS
cp-subscription-	A	0..1	The address (URI) of the "subscription manager"	URI	DS

Name	Type	Cardinality	Description	Data Type	Used By DCD Entities
manager-address			service of the Content Provider for the channel.		
notify-on-subscription-change	A	0..1	Indication that Content Provider should be informed when a user subscribes to the channel, or changes an existing subscription. Values: **0 – False (*)** 1 – True	Boolean	DS
proxied-subscription	A	0..1	Indication that the DCD Service Provider should handle subscription for this channel. Values: 0 – False **1- True (*)**	Boolean	DS
push-schedule	E2	0..N	Set of schedules upon which the DCD Server should push channel updates via DCD-2. This parameter is included in the metadata subset when the DCD Content Provider wants to influence network use and/or subscriber experience. Contains the following **attributes**: push-start-date push-start-time push-interval push-end-date push-end-time	Structure	DS
push-start-date	A	0..1	Date on which push publication begins.	Date	DS
push-start-time	A	0..1	Time at which interval push should begin. If the time value is a clock time only (does not include a weekday or date indication), push should begin daily at the given time.	Time	DS
push-interval	A	0..1	Anticipated interval between content update push publications by the Content Provider, in seconds.	Integer	DS
push-end-date	A	0..1	Date on which push publication ends	Date	DS
push-end-time	A	0..1	Time at which interval push ends. If the time value is a clock time only (does not include a weekday or date indication), push should end daily at the given time.	Time	DS

Table 162 Channel Publication Metadata

8.3 Content Metadata

8.3.1 Content Metadata

This section is specific to Content Metadata purpose, attributes, and relevance to DCD entities and interfaces. Other sections describe the requirements specific to DCD operations that include exchange and use of Content Metadata.

Content Metadata is defined in [DCD-AD] as: *"The Content Metadata is a set of dynamic settings and rules for handling delivery of the DCD Content. Content Metadata is associated with a particular unit of content and may include rules and settings for content expiry, replacement, fragmentation, etc."*

Content Metadata can be included in DCD content delivery interface request and responses, but is distinct from other request/response metadata, e.g. the common attributes of the DCD interface transactions (e.g. the Session ID) or values common to all content for particular channels (e.g. the Channel ID).

Following is an informative description of the source and submission of Content Metadata.

Content Metadata may originate from a variety of sources and have relevance to a variety of DCD entities, e.g.:

- From Content Providers, as part of content item publication to the DCD Server (pushed to the DCD Server for immediate or later delivery) or delivery in response to DCD Server requests (as part of a scheduled pull for content updates or as requested on-demand by DCD-Enabled Client Applications through DCD Clients). The relevance of

the metadata could be limited to the DCD Server (e.g. delivery rules for specific content items), to the DCD Client (e.g. delivery acknowledgement requirements), or could be relevant to all the entities including the DCD-Enabled Client Application (e.g. expiration time for content items).

- From the DCD Server, as part of its role in delivering specific content items, e.g. related to delivery/handling rules (e.g. expiration controls for storage management based upon available storage of a specific DCD Client) and features (e.g. compression for over-the-air efficiency) that the DCD Server may apply in delivering the content.

Note that earlier received content metadata may also be used to set request/response attributes from the DCD Client and DCD-Enabled Client Application:

- From the DCD Client, as part of requests or responses to the DCD Server, e.g. a content ID included in a delivery acknowledgement, or a request for on-demand delivery of a specific content item.
- From the DCD-Enabled Client Application, e.g. for on-demand delivery of a specific content item, e.g. the content is used by Application or not. If the shared memory is used by a browser, the DCD Client can't delete it.

Following is an informative description of Content Metadata attributes and elements. The mandatory/optional status and interface relevance of the attributes and elements is described in the table following.

- **Content ID**: identifies the content item. The content ID is set by the Content Provider, and is unique within the DCD Service Provider's domain. The main purpose of the content ID is to enable application level confirmation and resumption of content delivery.
- **Channel ID**: a list of channel IDs as assigned by the DCD Server. The Content Provider includes this attribute to associate content items with one or more DCD channels.
- **MIME type**: the MIME type of the content item.
- **Content types**: a list of strings that describe the channel content to enable association or filtering e.g. by "type", "category", "tag", or "relation". This is a comma-separated list of tokens of undefined grammar. Content Providers can use this attribute to indicate the specific types related to each content item.
- **Content length**: the size in bytes of the content item.
- **Content name**: name of content in a human readable format.
- **Replaces content ID**: content ID of an outdated content item that, if present in the storage of the DCD Server or DCD Client, should be replaced with this content item.
- **Content price**: indicates the price associated with this content item.
- **Content delivery notification**: indicates the need for the DCD Client to respond to the DCD Server with delivery acknowledgement for this content item via the DCD-1 interface.
- **Delivery Priority**: the priority associated with this content item. Note that DCD is only responsible for delivery of this attribute; presentation aspects are out of scope.
- **Content encoding**: encoding that has been applied to the content item, e.g. GZIP or deflate compression.
- **Content address**: an address (URL) where the content item can be directly retrieved by the DCD Client via the DCD-1 interface. If the content item is not present (i.e. embedded) in the current DCD operation, this address will be used by the DCD Client to retrieve the content item. This attribute can also be used by the DCD-Enabled Client Application to re-retrieve or request an update of the content item.
- **Content storage location**: the location of the content item in the DCD Client managed storage.
- **Content block ID**: identifier via which multiple content items can be associated as a block.
- **Parental Rating**: Content rating per FCC "TV Parental Guidelines" or similar local regulatory requirements.

- **Delivery rules**: attributes which control the delivery or retrieval of the content item:
    - **Deliver to:** a particular set of users to receive the content or the identifier specifying a group of users
    - **Deliver at:** absolute time at which the DCD Server should deliver a content item.
    - **Expires**: the expected lifetime of the content item in device storage, and the time until which the content item can be directly retrieved (if a content address attribute was provided). If the DCD Client is managing content storage for the channel, it will remove the item after that time.
    - **Delivery spread:** the period over which the DCD Server can randomize delivery, for the purpose of load spreading. A small or zero spread value can be used for time-sensitive content.
    - **Deliver when roaming:** indicates whether the content item should be automatically delivered or retrieved via point-to-point interfaces if the user is roaming. This content-item-specific attribute allows DCD channel updates to continue when roaming (deliver when roaming is allowed at the channel level), with on-demand retrieval only for specific content items. Thus the set of available content items for the channel can be known by the DCD-Enabled Client Application, for on-demand retrieval as requested by the user. Roaming-aware DCD Servers can use this attribute to determine whether to embed content items inside channel update responses over DCD-1, or to include the content address attribute only. Roaming-aware DCD Clients can use this attribute to determine whether to automatically retrieve content items which are referenced by content address only.
    - **Network preferences:** arbitrary criteria for network selection (e.g. GPRS vs. UMTS vs. Wi-Fi) based on delivery cost, bandwidth, quality of service, etc. DCD Client and DCD Server apply these criteria for content item delivery over DCD-1 and DCD-2 interfaces. Content Provider specifies this attribute based on the XML schema published by Service Provider (not in scope for DCD).
    - **Aux content link:** provides the content ID or link to the additional content that is related to the content being delivered. The intent is to support pre-fetching content referred to by the main content item or likely to be requested later.
- **Delivery personalization**: attributes which control personalization of the delivery or retrieval of the content item:
    - **Deliver per location**: An expression for matching a location at which delivery should be allowed
    - **Deliver per presence**: An expression for allowing delivery based upon matching a Presence attribute
    - **Deliver per XDMS**: An expression for allowing delivery based upon matching a XDMS attribute

8.3.2 Content Metadata Elements, Attributes, and Values

This section is defines the Content Metadata in detail.

Unless otherwise specified, all Content Metadata elements and attributes are mandatory for support by DCD Servers and DCD Clients, and optional for use.

Explanation of the Type in Table 163 Content Metadata:

- E = Elements
- A = Attributes
- 1,2,3, etc = nesting level of the element or attribute

Content Metadata basic structure:

content-metadata – E

- content-updated – A
- content-id – A

- channel-id – A
- mime-type – A
- content-types – A
- content-length – A
- content-name – A
- replaces-content-id – A
- content-price – A
- content-delivery-notification – A
- delivery-priority – A
- emergency-content – A
- content-encoding – A
- content-address – A
- content-storage-location – A
- content-block-id – A
- deliver-to – A
- deliver-at – A
- content-expiration – A
- delivery-spread – A
- deliver-when-roaming – A
- parental-rating – A
- network-preferences – A
- aux-content-link – A
- deliver-per-location – E1
- deliver-per-presence – E1
- deliver-per-xdms – E1

Name	Type	Cardinality	Description	Data Type	Used by	Originated from	Provided to DECA
Content-metadata	E	1	Contains the following **attributes**: content-updated content-id channel-id mime-type content-length content-types content-name replaces-content-id content-price content-delivery-notification delivery-priority emergency-content content-encoding content-address content-storage-location content-block-id parental-rating deliver-to deliver-at content-expiration delivery-spread deliver-when-roaming network-preferences aux-content-link Contains the following **sub-elements** deliver-per-location deliver-per-presence deliver-per-xdms	Structure	DS, DC	CP	YES
Content-updated	A	0..1	Time when the content item was last updated. SHALL conform to the "date-time" definition in [ISO8601]. In addition, an uppercase "T" character	dateTime	DC	CP, DS	YES

Name	Type	Cardinality	Description	Data Type	Used by	Originated from	Provided to DECA
			SHALL be used to separate date and time, and an uppercase "Z" character SHALL be present.				
Content-id	A	1	Identifier set by the Content Provider, and unique within the DCD Service Provider's domain. The main purpose of the content ID is to enable application level confirmation and resumption of content delivery. Implementation in XML schema will use the "AnyURI" data type.	String	DS, DC	CP	YES
Channel-id	A	0..1	A list of Channel IDs as assigned by the DCD Server. The Content Provider includes this attribute to associate content items with DCD channels	List of Strings	DS, DC	CP	YES
mime-type	A	1	The MIME type of the content item.	String	DS, DC	CP	YES
Content-length	A	1	The size in bytes of the content item.	Integer	DS, DC	CP, DS	YES
Content-types	A	0..1	A list of strings that describe the channel content to enable association or filtering e.g. by "type", "category", "tag", or "relation"	List of Strings	DS, DC	CP	YES
Content-name	A	0..1	Name of content in a human readable format.	String	DC	CP	YES
replaces-content-id	A	0..1	Content ID of an outdated content item that, if present in the storage of the DCD Server or DCD Client, should be replaced with this content item. Implementation in XML schema will use the "AnyURI" data type.	String	DS, DC	CP	YES
Content-price	A	0..1	Indicates the price (amount and currency) of this content item. The purpose is to let the user know the price of the content and decide if he wants to retrieve it	String	DC	CP, DS	YES
Content-delivery-notification	A	0..1	Indicates the need for, or status of, delivery acknowledgement for this content item. This attribute should be set to "true" if user is charged for delivery of this content item (subject to DCD Service Provider Policy) Values: **0 – False (*)** 1 – True	Boolean	DS, DC	CP, DS	NO
Delivery-priority	A	0..1	The delivery priority associated with this content item. Values: 1 – Low 2 – Medium (*) 3 – High	Enumerated	DS, DC	CP, DS	NO
emergency-content	A	0..1	Indicates whether the content item contains emergency content. Values: **0 – False (*)** 1 – True	Boolean	DC	CP	YES
Content-encoding	A	0..1	Encoding that has been applied to the content item, e.g. GZIP or deflate compression.	String	DC	DS	NO
Content-address	A	0..1	An address (URL) where the content item can be directly retrieved by the DCD Client via the DCD-1 interface.	URI	DC	DS	YES
Content-storage-location	A	0..1	Location of the content item in the DCD Client managed storage	String	DC	DC	YES
Content-block-id	A	0..1	Identifies which multiple content items can be associated as a block. May be used by the DCD Server for content aggregation / bundling.	String	DS	CP	NO
parental-rating	A	0..1	Content rating per FCC "TV Parental Guidelines" or similar local regulatory requirements. May be used by the DCD Server for content selection / filtering, and by DCD Client for the same in the broadcast case.	String	DS, DC	CP	YES
deliver-to	A	0..1	A particular set of users to receive the content.	String	DS	CP	NO
deliver-at	A	0..1	Time at which the DCD Server should deliver a content item. 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.	dateTime	DS	CP	NO
Content-expiration	A	0..1	The expected lifetime of the content item in device storage, and the time until which the content item can be directly retrieved (if a content address attribute	String	DS, DC	CP	YES

Name	Type	Cardinality	Description	Data Type	Used by	Originated from	Provided to DECA
			was provided).				
Delivery-spread	A	0..1	The period over which the DCD Server can randomize delivery, for the purpose of load spreading.	String	DS	CP	NO
deliver-when-roaming	A	0..1	Indicates whether the content item should be automatically delivered or retrieved via point-to-point interfaces if the user is roaming Values: **0** – **False (*)** 1 – True	Boolean	DS	CP	NO
Network-preferences	A	0..1	Descending priority-ordered, comma-separated list of network/bearer types for use in content delivery, selected per arbitrary deployment-specific criteria for network selection (e.g. GPRS vs. UMTS vs. Wi-Fi) based on delivery cost, bandwidth, quality of service, etc. DCD Client and DCD Server apply these criteria for content delivery over DCD-1 and DCD-2 interfaces. One or more of “EDGE”, “UMTS”, “CDMA2000”, “GAN”, “WiMAX”, “LTE”, “802.11”, “CBS”, “BCAST”.	String	CP, DS,	DS	NO
aux-content-link	A	0..1	Provides link to additional content that is related to the content being delivered. The intent is to support pre-fetching content referred to by the main content item or likely to be requested later.	URI	DS, DC	CP, DS	NO
deliver-per-location	E1	0..1	A rule for matching a location at which delivery should be allowed. The rule should be specified by the DCD Content Provider, the DCD Server, or the DCD Enabled Client Application based on the format published by the Service Provider	String	DS	CP	NO
deliver-per-presence	E1	0..1	A rule for allowing delivery based upon matching a Presence attribute	String	DS	CP	NO
deliver-per-xdms	E1	0..1	A rule for allowing delivery based upon matching a XDMS attribute.	String	DS	CP	NO

Table 163 Content Metadata

Table Notes:

(1) The flow of channel metadata is from the Content Provider in the direction of the DCD-Enabled Client Application (“downstream”). Some metadata items may originate in the DCD Server or DCD Client.

(2) “Conditional” means if that the attribute is present in the previous “upstream” interface, then it SHALL be present in the interface as shown.

(*) Default value to use when the attribute is omitted

9. DCD Content Packaging

9.1 Logical Model

The DCD Content Packaging format is based on the logical model described in the Section 5.3 of [DCD-AD].

When the DCD Content Packaging format is used by the DCD Content Provider, the DCD Content received by the DCD Server is packaged along with the DCD Content Metadata and wrapped in the DCD envelope. The DCD Server extracts a Content Metadata subset related to server-side processing from the DCD envelope, uses this metadata for content processing, and sends a client DCD envelope to the DCD Client. Alternatively, the DCD Content and client-related Content Metadata may be packaged into a client DCD envelope by the DCD Server. The client DCD envelope contains the DCD Content and Content Metadata for the DCD Client. The DCD Client extracts Content Metadata, uses this metadata for content processing, and forwards the DCD Content to the DCD Enabled Client Application or stores it for retrieval by the application.

When the DCD Content is of syndicated format such as an ATOM or RSS feed, the DCD Content Provider can embed the DCD Content Metadata directly into the feed (see Sections 9.2 and 9.3 for details) and provide this ATOM or RSS feed to the DCD Server. The DCD Server extracts a Content Metadata subset related to server-side processing from the ATOM or RSS feed received from the DCD Content Provider, uses this metadata for content processing, and sends the resulting ATOM or RSS feed to the DCD Client, if this DCD Client supports ATOM or RSS feed format over DCD-1/2 interfaces. Otherwise, the DCD Server repackages the feed and sends it to the DCD Client in the DCD Content Packaging format. The DCD Client extracts Content Metadata and forwards the ATOM or RSS feed to the DCD Enabled Client Application (or stores it for retrieval by the application).

DCD Client support for ATOM or RSS feed format over DCD-1/2 interfaces is determined by the DCD Server during session activation, through the DCD Client's User Agent Profile (UAProf). This may be the UAProf of the device referencing the embedded DCD Client's capabilities, or a UAProf of a downloaded DCD Client.

When the DCD Content is of syndicated format, it is preferred that the DCD Content Provider uses the aforementioned packaging method with Content Metadata embedded directly in the ATOM or RSS feed (see Sections 9.2 and 9.3).

DCD content packaging requirements are defined as follows:

1. Support requirements:

DCD Server SHALL support content publishing in the following formats:

- DCD Content Packaging format with the Content Metadata embedded along with the DCD Content in the DCD envelope
- ATOM Feed Packaging format with the Content Metadata embedded after the <feed> XML element that contains DCD namespace reference.
- RSS Feed Packaging format with the Content Metadata embedded after the <rss> XML element that contains DCD namespace reference.

DCD Server SHALL support DCD Content Packaging format over DCD-1/2 interfaces

DCD Server SHALL support content packaging with Content Metadata embedded in the feed for delivery of ATOM or RSS content over DCD-1/2 interfaces

DCD Server SHALL support repackaging of ATOM and RSS feeds into the DCD Content Packaging format.

DCD Client SHALL support DCD Content Packaging format over DCD-1/2 interfaces

DCD Client MAY support content packaging with Content Metadata embedded in the feed for delivery of ATOM and RSS content over DCD-1/2 interfaces

DCD Client SHALL support forwarding of syndicated formatted (ATOM or RSS) content received from the DCD Server over DCD 1/2 interfaces to the DCD Enabled Client Application

DCD Client SHALL allow the DCD Enabled Client Application to use any content format over DCD-CADE in an opaque manner.

DCD Client SHALL support DCD XML schema over DCD-CADE, DCD-CAR, DCD-1/2/3

DCD Server SHALL support DCD XML schema over DCD-CPDE, DCD-CPR, DCD-1/2/3

2. Use requirements:

DCD Server SHALL use DCD Content Packaging format for delivery of non-syndicated content over DCD-1/2 interfaces

DCD Server SHALL use DCD Content Packaging format for delivery of ATOM or RSS content over DCD-1/2 interfaces, if the DCD Client doesn't support content packaging with Content Metadata embedded in the feed over DCD-1/2 interfaces. In this case the DCD Server SHALL repackage ATOM and RSS content provided by the DCD Content Provider into the DCD envelope

DCD Server SHALL use content packaging with Content Metadata embedded in the feed for delivery of ATOM or RSS content over DCD-1/2 interfaces if the DCD Client supports content packaging by inserting Content Metadata into the feed over DCD-1/2 interfaces. In this case the DCD Server SHALL extract appropriate server-side related Content Metadata from the feed received from the DCD Content Provider and forward the rest of the feed to the DCD Client.

If the content is delivered over DCD-1/2 interfaces using DCD Content Packaging format, the DCD Client SHALL extract Content Metadata and DCD Content from the DCD envelope and pass the DCD Content to the DCD Enabled Client Application according to the appropriate settings in Channel and Content Metadata.

If ATOM or RSS content is delivered over DCD-1/2 interfaces using content packaging by inserting Content Metadata into the feed, the DCD Client SHALL extract Content Metadata from the feed and pass the feed to the DCD Enabled Client Application according to the appropriate settings in Channel and Content Metadata.

```
<?xml version="1.0"?>
<server-package xmlns="http://www.openmobilealliance.com/oma-dcd/1.0>
    <ds-cnt-metadata content-block-id="0xA017" content-id="0xB12F" deliver-at="2009-01-10T18:30:02Z"
                    deliver-when-roaming=true network-preferences="WiMAX;UMTS;LTE;802.11;"
                    parental-rating="G" delivery-priority="1 - Low" deliver-per-presence="willingness=open" …/>
    <client-package>
      <dc-cnt-metadata content-price="0.02" content-id="0xB12F" content-name="Hot News" parental-rating="G"
                       delivery-priority="1 – Low" …/>
      <content>
          <!– the content block is opaque for the DCD enabler and should be parsed by the application only -->
          <x:meta attribute1="Vive la DCD!" attribute2="some other metadata"/>
          <x:text>"Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings during
          its key holiday quarter, topping analysts' estimates. Shares of Wal-Mart jumped nearly 3 percent"
          </x:text>
          <x:image>"
          WpjQu5yqmCYsapyuvUUlvONmOZtfzgFzfagowXhGargtWNO7PsFk90xQpqPTOrNTZ0Ux9xdvljLpwz3"
          </x:image>
          <x:ad>http://myads.com/SSXLSW&dc</x:ad>
       </content>
    </client-package>
</server-package>
```

Figure 74: DCD Content Packaging Format Example

Figure 74: DCD Content Packaging Format Example shows an example of DCD Content as received from the DCD Content Provider and packaged according to the DCD Content Packaging format (see Section 9.4 for details on DCD XML schema).

9.2 RSS Extensions

DCD enabler supports RSS and allows extending the RSS XML schema with DCD extensions for embedding DCD Content Metadata inside the RSS feed. The DCD Content Provider embeds DCD Content Metadata into the feed to facilitate content processing by the DCD enabler. The DCD Server extracts the Content Metadata related to server operations, and depending on whether the target DCD Client supports RSS packaging over DCD-1/2 interfaces:

- forwards the RSS feed with client-related Content Metadata to the DCD Client, or
- repackages the RSS feed into DCD Content Packaging format and forwards it to the DCD Client

In the event that the DCD Content Provider does not embed DCD Content Metadata in the RSS-formatted DCD Content, the DCD Server SHALL use default values for Content Metadata and embed the client-related part of the Content Metadata when it forwards the DCD Content to the DCD Client. The DCD Server MAY use the metadata in the RSS feed to create default metadata for the content item.

To maintain the opacity of DCD Content published in the RSS format, the DCD Content Metadata SHALL be embedded into the RSS feed before the content payload i.e. immediately after the <channel> XML tag. The xmlns attribute with a DCD namespace reference SHALL be included in the <rss> (or <rdf>) XML element. The <rss> (or <rdf>) XML element MAY also include a schemaLocation attribute specifying the location of the DCD XML schema document.

When the DCD Client delivers the RSS-formatted DCD Content to the DCD Enabled Client Application, the DCD Client SHALL remove from the RSS feed all DCD Content Metadata as well as the DCD namespace reference attribute and the DCD XML schema location attribute, if included.

Figure 75: Example of RSS-formatted DCD Content with embedded DCD Content Metadata shows an example of RSS-formatted DCD Content, as received from the DCD Content Provider (see Section 9.4 for details on DCD XML schema).

```
<?xml version="1.0" ?>
<rss version="2.0" xmlns:dcd="http://www.openmobilealliance.com/oma-dcd/1.0>
    <channel>
        <dcd:ds-cnt-metadata content-block-id="0xA017" content-id="0xB12F" deliver-at="2009-1-10T18:30:02Z"
            deliver-when-roaming=true  network-preferences="802.11;WiMAX;LTE;UMTS"
            parental-rating='G' delivery-priority='1' deliver-per-presence="willingness=open"…/>

        <dcd:dc-cnt-metadata content-price="0.02" content-id="0xB12F" parental-rating="G" delivery-priority='1'
            content-name="Hot News" … />
<!– the content below is opaque for the DCD enabler and should be parsed by the application only -->
        <title>Top business stories</title>
        <link>http://www.abc.com/bus/rss</link>
        <description> Top business stories</description>
        <item>
            <title>Wal-Mart reports jump in profit</title>
            <link>http://abc.com/click/here.pl?80da344efa6a</link>
            <pubDate>Sat, 10 Jan 2009 12:07:31 GMT</pubDate>
            <description>Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings
                during its key holiday quarter, topping analysts' estimates.Shares of Wal-Mart jumped
                nearly 3 percent
            </description>
            <guid>http://abc.com/id?80da344efa6a</guid>
        </item>
        <image>
            <url>http://abc.com.au/top/img/wmart.gif</url>
            <title>Wal-Mart beats estimates</title>
            <link>http://www.abc.com/bus/rss</link>
            <width>142</width>
            <height>40</height>
        </image>
    </channel>
</rss>
```

Figure 75: Example of RSS-formatted DCD Content with embedded DCD Content Metadata

In case of RSS based packaging format, message semantics are embedded inside the feed – see examples in Appendix C2

9.3 Atom Extensions

DCD enabler supports ATOM [RFC 4287] and allows extending the ATOM XML schema with DCD extensions for embedding DCD Content Metadata inside the ATOM feed. The DCD Content Provider embeds DCD Content Metadata into the feed to facilitate content processing by the DCD enabler. The DCD Server extracts the Content Metadata related to server operations, and depending on whether the target DCD Client supports ATOM packaging over DCD-1/2 interfaces:

- forwards the ATOM feed with client-related Content Metadata to the DCD Client, or
- repackages the ATOM feed into DCD Content Packaging format and forwards it to the DCD Client

In the event that the DCD Content Provider does not embed DCD Content Metadata in the ATOM-formatted DCD Content, the DCD Server SHALL use default values for Content Metadata and embed the client-related part of the Content Metadata

when it forwards the DCD Content to the DCD Client. The DCD Server MAY use the metadata in the ATOM feed to create default metadata for the content item.

To maintain the opacity of DCD Content published in the ATOM format, the DCD Content Metadata SHALL be included after the <feed> XML tag. The xmlns attribute with a DCD namespace reference SHALL be included in the <feed> XML element. The <feed> XML element MAY also include a schemaLocation attribute specifying the location of the DCD XML schema document.

When the DCD Client delivers the ATOM-formatted DCD Content to the DCD Enabled Client Application, the DCD Client SHALL remove from the ATOM feed all DCD Content Metadata as well as the DCD namespace reference attribute and the DCD XML schema location attribute, if included.

Figure 76: Example of ATOM-formatted DCD Content with embedded DCD Content Metadata shows an example of ATOM-formatted DCD Content, as received from the DCD Content Provider (see Section 9.4 for details on DCD XML schema).

```
<?xml version="1.0" ?>
<feed xmlns=http://www.w3.org/2005/Atom xmlns:dcd="http://www.openmobilealliance.com/oma-dcd/1.0>

    <dcd:ds-cnt-metadata content-block-id="0xA017" content-id="0xB12F" deliver-at="2009-01-10T18:30:02Z"
                         deliver-when-roaming=true network-preferences="WiMAX;UMTS;LTE;802.11;"
                         parental-rating="G" delivery-priority="1 - Low" deliver-per-presence="willingness=open" …/>

    <dcd:dc-cnt-metadata content-price="0.02" content-id="0xB12F" content-name="Hot News" parental-rating="G"
                         delivery-priority="1 – Low" …/>

<!– the content below is opaque for the DCD enabler and should be parsed by the application only -->
    <title>Top business stories</title>
    <link href="http://www.abc.com/bus/rss/">
    <updated>2009-01-08T11:59:55Z</updated>
    <id>urn:uuid:1724c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <author>
        <name>James Willis</name>
        <email>jwillis@abc.com</email>
    </author>
    <entry>
        <title>Wal-Mart reports jump in profit</title>
        <summary>Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings
                 during its key holiday quarter, topping analysts' estimates.Shares of Wal-Mart jumped
                 nearly 3 percent
        </summary>
        <id>urn:uuid:61a73c80-d399-11d9-b93C-0003939e0af6</id>
        <updated>2007-02-21T11:59:55Z</updated>
        <link>http://abc.com/click/here.pl?80da344efa6a</link>
    </entry>
    <entry>
     …
    </entry>
</feed>
```

Figure 76: Example of ATOM-formatted DCD Content with embedded DCD Content Metadata

In case of ATOM based packaging format, message semantics are embedded inside the feed – see examples in Appendix C2

9.4 DCD XML Schema

The DCD XML schema described in this section is based on the definitions of DCD messages (Section 7), Application Profile (Section 8.1), Channel Metadata (Section 8.2), Content Metadata (Section 8.3) and DCD packaging model (Section 9.1).

The XML schema defines DCD messages as distinct XML elements (e.g. ChannelDiscoveryConfirmation element) with message structure enforced according to the definitions in Section 7. This approach ensures better interoperability, built-in message validation, and parsing efficiency.

DCD content packaging is enforced through the use of Server-Package, Client-Package, and DCD-Content as valid types for the Content-Package element. These types contain the appropriate metadata and the nested content package. For example, the Server-Package is required to contain DS-Content-Metadata and the Client-Package. If the content is published as ATOM or RSS feed, the publisher can embed DCD Content Metadata (i.e. DS-Content-Metadata and DC-Content-Metadata) directly after the line with namespace reference to the DCD XML schema.

See [DCD-XSDDCD-XSDDCD-XSDDCD-XSD] for XML schema definition.

10.DCD Security

10.1 Authentication

10.1.1 Authentication in Session Establishment

In session establishment, DCD Clients act as subscriber agents for authentication purposes, and DCD Servers act as agents for DCD Service Providers for authentication purposes. DCD Service Providers act upon the channel security policy required by Content Providers. This section refers to authentication of the DCD Client and DCD Servers acting in those roles.

Authentication during session establishment ensures that DCD Clients and DCD Servers can trust the other entity in the session relationship. The necessity of establishing this trust relationship may vary for different DCD Service Providers or DCD Clients. This necessity is assumed to be known through preconfiguration, or through discovery as explicitly supported by the mechanisms described below. Other discovery mechanisms are unspecified.

DCD Clients SHALL supply authentication in session establishment, per the security policy of the DCD Server with which a session is being established. DCD Servers SHALL supply authentication in session establishment, per the security policy of the DCD Client with which a session is being established. DCD Clients and DCD Servers SHALL accept valid authentication, when supplied per their security policy.

Three modes of DCD Client authentication are specified for use by the DCD enabler:

- DCD Client authentication provided by functions external to the DCD enabler. DCD Servers SHALL support this method of DCD Client authentication.
- DCD Client authentication via HTTP Digest Authentication [RFC2617]. DCD Servers SHALL support this method of DCD Client authentication. DCD Clients SHALL support this method of DCD Client authentication.
- DCD Client authentication via a lower-layer security facility, e.g. TLS-based delivery of a client certificate. DCD Servers SHOULD support this method of DCD Client authentication. DCD Clients SHOULD support this method of DCD Client authentication.

If DCD Client authentication is not recognized or otherwise not accepted by a DCD Server (e.g. the specific subscriber is not authorized to activate the DCD enabler as requested), the DCD Server SHALL provide an “authentication error” response to the DCD Client. Upon an “authentication error” response during session establishment, DCD Clients SHALL provide an appropriate error response to affected DCD-Enabled Client Applications. Further DCD Client behavior in this case (e.g. session establishment retry) is unspecified.

One mode of DCD Server authentication is specified for use by the DCD enabler:

- DCD Server credentials-based authentication via a lower-layer security facility, e.g. delivery of a server certificate. DCD Servers SHALL support this method of DCD Server authentication. DCD Clients SHALL support this method of DCD Server authentication.

If DCD Server authentication is not recognized or otherwise not accepted by a DCD Client (e.g. the server certificate has expired or cannot be validated), the DCD Client SHALL provide an “authentication error” response to the DCD Server. Further DCD Server behavior in this case is unspecified. Upon an “authentication error” response during session establishment, DCD Clients SHALL provide an appropriate error response to affected DCD-Enabled Client Applications. Further DCD Client behavior in this case (e.g. session establishment retry) is unspecified.

In addition, unauthenticated service may be provided if allowed per the security policy of the DCD Server and/or DCD Client. DCD Servers SHALL provide session establishment without DCD Client authentication, if allowed per DCD Service Provider security policy. DCD Clients SHALL provide session establishment without DCD Server authentication, if allowed per DCD Client security policy.

10.1.1.1 DCD Client Authentication provided by functions external to the DCD enabler

In this mode, DCD Client authentication is provided by non-DCD network elements, e.g. as operated by the DCD Service Provider. This is a typical approach to client authentication in cases which client-provided authentication is not available or not trusted. The details are unspecified, but may include such options as:

- Subscriber identification via RADIUS, provided directly to the DCD Server
- Proxy-inserted subscriber identification, e.g.
    - The DCD Client is configured to use a proxy for the DCD-3 interface, and sends DCD-3 requests via the proxy instead of direct to the DCD Server.
    - The proxy inserts a subscriber identity indication (e.g. HTTP header) based upon network information (e.g. from RADIUS) into the request, and forwards it to the DCD Server. The subscriber identity may be in any form supported/permitted to be inserted by the network proxy for delivery to all or specific destinations.

10.1.1.2 DCD Client authentication via HTTP Digest Authentication

In this mode, DCD Client authentication is provided by the DCD Client from pre-configured or subscriber-provided credentials. The mechanisms for pre-configuration or subscriber entry of credentials are unspecified.

HTTP Digest Authentication [RFC2617] is used to deliver the credentials to the DCD Server:

1. The DCD Client sends a ClientActivationRequest message to the DCD Server. If the DCD Client supports Digest Authentication per the GBA Profile of [SEC_CF], it discloses this as described in [TS24109].
2. The DCD Server indicates the need for DCD Client authentication via Digest Authentication by responding to the first DCD-3 interface message with a digest access authentication challenge, including a generated nonce value. If the DCD Server supports Digest Authentication per the GBA Profile of [SEC_CF], it authenticates the UE using GBA-based shared secret as described in [TS24109], per the policy established by the DCD Service Provider.
3. The DCD Client sends a challenge response with its credentials, hashed per the supplied nonce value. Note the credentials may be derived from a variety of sources, e.g. user-provided/stored or provisioned “username:password”, or 3GPP GBA-provided credentials as described for the GBA Profile of [SEC_CF].
4. The DCD Server retrieves the credentials from the hashed digest-response value.

10.1.1.3 DCD Client and DCD Server authentication via lower-layer security facilities

In this mode, DCD Client and/or DCD Server authentication is provided by facilities of an underlying security layer, e.g. TLS-provided client or server certificates. The methods of certificate management and notification of error conditions to the DCD Client or DCD Server are assumed to be addressed by the underlying transport security, and are unspecified.

10.1.2 Client Authentication for Messages in DCD-1, DCD-2 and DCD-3 Interfaces

Once DCD Client authentication has been established during session activation, authentication for DCD-1 and DCD-3 messages, if needed, can be based upon the same methods described in Section 10.1.1. The authentication method to be used is per the policy of the DCD Service Provider.

DCD Servers and DCD Clients SHALL support use of the Session ID as an authentication token in DCD-1 and DCD-3 messages.

DCD Servers and DCD Clients SHALL support use of the Session ID as an authentication token in DCD-2 messages directed at a specific DCD Client over point-to-point transports.

If session establishment included HTTP Digest Authentication per the GBA Profile of [SEC_CF], DCD Servers and DCD Clients SHALL use the GBA-derived Bootstrapping Transaction Identifier (B-TID) to set or validate the Session ID in messages over the DCD-1 interface, DCD-3 interface, and point-to-point DCD-2 interface.

10.1.3 Reauthentication upon Session Timeout

Upon session expiration, due to Session TTL being reached or the expiration of the GBA B-TID as Session ID, the DCD Client SHALL re-activate the session via the same procedure described in Section 10.1.1, including reauthentication as necessary.

10.2 Connection Security

10.2.1 Connection Security in Session Establishment

Connection security during session establishment ensures that DCD Clients and DCD Servers can trust the privacy and integrity of the operations and data exchanged with the other entity during the session establishment. The necessity of connection security during session establishment may vary for different DCD Service Providers or DCD Clients. This necessity is assumed to be known through preconfiguration, or through discovery as explicitly supported by the mechanisms described below. Other discovery mechanisms are unspecified.

DCD Clients SHALL provide connection security for session establishment, per the security policy of the DCD Server with which a session is being established. DCD Servers SHALL provide connection security for session establishment, per the security policy of the DCD Client with which a session is being established.

One mode of connection security for session establishment is specified for use by the DCD enabler:

- Use of transport layer security provided by TLS [RFC4346] or DTLS [RFC4347]. DCD Servers and DCD Clients SHALL support TLS or DTLS to provide connection security for DCD-3 interface operations and data, as applicable to the underlying transport protocol in use for DCD-3.

11.DCD in Broadcast Realm

DCD transactions initiated from the DCD Server and carried by broadcast bearer (see table in Section 5.9) which require response (according to the transaction flows in Section 7) may imply overload on the DCD Server due to simultaneously incoming messages from all broadcast enabled DCD Clients.

In order to spread over time the resulting load the DCD Client SHALL take measures to ensure that responses to DCD Server are distributed over time (for example, a random delay based on the response-spread attribute in the Application Profile if exist or any other arbitrary value for spreading).

When DCD Client interaction with the DCD Server is required in response to DCD-2 or DCD-3 interface messages, this interaction occurs over point-to-point uplink bearers, using HTTP as the transport as described in Section 12.2.1.

When the pull DCD-3 interface is unavailable because the uplink is unavailable, DCD Client SHOULD continue to support the DCD-3 interface via Push or broadcast bearers, if available and feasible.

When the DCD-1interface is unavailable because the uplink is unavailable, DCD Client SHOULD continue to support reception of content via the DCD-2 interface for subscribed channels, if feasible

11.1 BCAST

DCD entities operating over the OMA BCAST enabler SHALL comply with [DCD-TS-BCAST] in addition to this specification.

11.2 Cell Broadcast

CBS is a unidirectional broadcast mechanism which is used as transport mechanism between DCD Server in the network and a DCD Client on the mobile device.

[3GPP 23.041] specify the cell broadcast basic structure for both GSM and UMTS.

DCD entities operating over the 3GPP CBS enabler SHALL comply with [DCD-TS-CBS] in addition to this specification.

The DCD Data part of the message when using CBS will contain the relevant DCD operations; those operations SHALL support both encoding rules of WBXML and/or of text/XML.

12.DCD in Point-to-Point Realm

12.1 Push Mode

The DCD Server SHOULD support either the Push Access Protocol (PAP) [PUSH-PAP] or the Push-OTA protocol [PUSH-OTA] for point-to-point Content Push via the DCD-2 interface, and point-to-point system message Push via the DCD-3 interface.

For DCD Push messages, the DCD Server SHALL include the Push Application ID header "X-Wap-Application-Id: x-wap-application:dcd.ua".

When using PAP, the DCD Server SHALL submit DCD messages for delivery using the MIME content type "application/vnd.oma.dcd".

When delivering DCD messages via Push-OTA, the DCD Server SHALL encode the messages using the MIME content type "application/vnd.oma.dcdc".

Push Clients in DCD supporting terminals SHALL support routing of Push messages with the Push Application ID header "X-Wap-Application-Id: x-wap-application:dcd.ua" to the DCD Client.

12.2 Pull Mode

12.2.1 HTTP Transport Binding

12.2.1.1 DCD-1, DCD-2, and DCD-3 Interfaces

DCD Clients and DCD Servers SHALL support Hypertext Transfer Protocol version 1.1 (HTTP [RFC2616]) for the DCD-1. DCD-2, and DCD-3 interfaces.

DCD Servers and DCD Clients SHALL support DCD-1, DCD-2, and DCD-3 interface messages formatted as entity-bodies with the application/vnd.oma.dcd media type. The application/vnd.oma.dcd media type is used when a single DCD interface message is included in the HTTP request/response.

DCD Servers and DCD Clients SHALL support DCD-1, DCD-2, and DCD-3 interface messages formatted as entity-bodies with the multipart/related media type. The multipart/related media type is used when DCD interface messages are delivered as several inter-related body parts in a single HTTP request/response.

DCD Servers and DCD Clients SHALL support DCD-1, DCD-2, and DCD-3 interface messages formatted as entity-bodies with the multipart/mixed media type. The multipart/mixed media type is used when multiple DCD interface messages are concatenated in a single HTTP request/response. Note that DCD Clients can only use the multipart/mixed media type for message concatenation if the same DCD Server address and network preferences applies to all the messages.

DCD Clients SHALL send all DCD-1 and DCD-3 interface messages as HTTP POST method requests, including:

- the DCD Server address in the request line, per the applicable connection profile for the DCD interface
- the Host request-header set to the hostname or IP address of the DCD Server, per the applicable connection profile for the DCD interface
- the User-Agent request-header set to identify the host device (e.g. "vendor-model/version"), and the name and version of the DCD Client as user agent initiating the request
- the Accept request-header with value "application/vnd.oma.dcd, multipart/related, multipart/mixed"
- the X-Wap-Profile request-header set to the URI of the User Agent Profile for the host device
- the Accept-Encoding request-header with value per the supported HTTP compression encodings, i.e. deflate and / or gzip

- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value “application/vnd.oma.dcd”, “multipart/related”, or “multipart/mixed”, as applicable
- the DCD-1 or DCD-3 message(s) as message-body

DCD Servers SHALL send all DCD-1, DCD-2, and DCD-3 interface messages sent as responses to DCD Client messages as the entity-body of HTTP 200 OK responses, including:

- the ETag entity-header set to a unique value within the scope of the DCD Server
- the Content-Encoding entity-header set to the type of HTTP compression applied, if any
- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value “application/vnd.oma.dcd”, “multipart/related”, or “multipart/mixed”, as applicable
- the DCD-1, DCD-2, or DCD-3 message(s) as message-body

DCD Servers SHALL NOT bundle multiple unrelated messages in an HTTP response for DCD Clients that support a Push bearer for the DCD-2 interface.

When there is no DCD message to send in response to a DCD Client request, the DCD Server SHALL send a 204 No Content response.

The following table provides an overview of the HTTP binding for each DCD-1, DCD-2, and DCD-3 transaction.

HTTP Method	Request Body	Response Code	Response Body
POST	ContentUpdateRequest	200 OK	ContentUpdateResponse
POST	ContentDeliveryConfirmation	204 No Content	
POST	ContentSubmitRequest	200 OK	One of: ContentUpdateResponse ContentSubmitConfirmation
POST	ClientActivationRequest	200 OK	ClientActivationResponse
POST	ClientDeactivationRequest	200 OK	ClientDeactivationResponse
POST	ApplicationRegistrationRequest	200 OK	ApplicationRegistrationResponse
POST	ApplicationDeregistrationConfirmation	204 No Content	
POST	ContexualInformationUpload	204 No Content	
POST	UsageTrackingReport	204 No Content	
POST	ChannelSubscriptionRequest	200 OK	ChannelSubscriptionResponse
POST	ChannelUnsubscriptionRequest	200 OK	ChannelUnsubscriptionResponse
POST	ChannelDiscoveryConfirmation	204 No Content	
POST	ChannelDiscoveryRequest	204 No Content	
POST	ChannelSuspendRequest	200 OK	ChannelSuspendResponse
POST	ChannelResumeRequest	200 OK	ChannelResumeResponse
POST	ContentRepairRequest	200 OK	ContentRepairResponse
POST	ChannelMetadataUpdateConfirmation	204 No Content	

Table 164 Message Relations between HTTP request and responses

HTTP Method	Request Body	Response Code	Response Body
POST	Any request message	200 OK	One or more of: ContentUpdatePush ContentUpdateNotification RequestForClientActivation ClientDeactivationNotification ApplicationDeregistrationNotification ContexualInformationUploadRequest RequestForUsageTrackingReport ChannelUnsubscriptionNotification SubscriptionNotification ChannelDiscoveryInfo ChannelDiscoveryNotification ChannelResumeNotification ChannelSuspendNotification ChannelMetadataUpdate

Table 165 Bundling Messages in HTTP response

12.2.1.2 Partial Messages Delivery

To support the capability of pause and resume for DCD messages, DCD Clients and DCD Servers SHALL support HTTP Partial GET requests (i.e. HTTP GET requests with *Range* header as defined in [RFC2616]).

The HTTP-level message flow of DCD Client initiated transactions is illustrated in "Figure 77: Message flow for initial request".

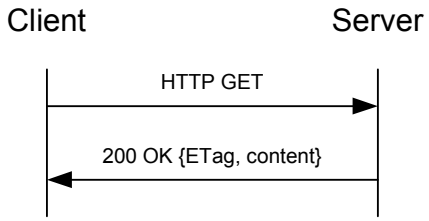


Figure 77: Message flow for initial request

If DCD Server response becomes interrupted before completion, the DCD Client SHALL NOT discard the received message part. The received message part SHALL be stored by the DCD Client but not made available to DCD-Enabled Client Application until the remaining part of the message is successfully received.

To receive the remaining part of the message, the DCD Client SHALL send a HTTP Partial GET request to the DCD Server. The policy (e.g. time interval and maximum amount of retries) between the download resumptions is implementation-specific. The request SHALL be sent to the same URI as the original message request. The DCD Client SHALL also include the following headers in the request:

- The *If-Match* header SHALL contain the value of the ETag header provided in the original download attempt of the content file.

- The *Range* header SHALL specify the remaining content range of the content file that has not yet been successfully received in the DCD Client.

Example: The DCD client has received the bytes 0-1233 in the initial download response. Thus it includes the following Range header in the Partial GET request:

Range: bytes 1234-

If the same message variant is still available in DCD Server (i.e. the ETag value included in the If-Match header matches with the ETag value of the message available in the DCD Server), the DCD Server responds with *206 Partial Content* message including the remaining message part in the message body. In this case the DCD Client SHALL concatenate the received message parts and further process the completed message.

The message flow of a resumed message is illustrated in "Figure 78: Message flow for resumed message".

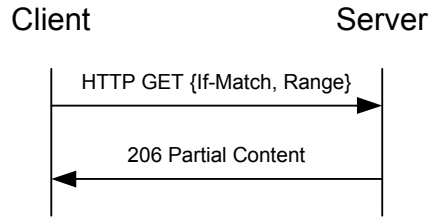


Figure 78: Message flow for resumed message

However, if the same message variant is no longer available in the DCD Server upon message resumption attempt, the DCD Server SHALL respond with an error code (i.e. *HTTP 412 Precondition Failed* if the ETag values do not match, *HTTP 410 Gone* or *HTTP 404 File Not Found* if the requested message is no longer available in the requested URI). When receiving any of these three error codes, the DCD Client SHALL cancel the message retry and not initiate any further retry attempts for this message. Further, the stored message part SHALL be discarded.

12.2.1.3 Messages and Information Elements

Message	Implementation	Direction
HTTP POST	Mandatory	DCD Client → DCD Server
Response to HTTP POST	Mandatory	DCD Client ←DCD Server
HTTP GET (for partial message delivery)	Mandatory	DCD Client → DCD Server
Response to HTTP GET	Mandatory	DCD Client ←DCD Server

Table 166 Message directions for HTTP GET/POST and responses between DCD Client and DCD Server

Information Element	Req	Type	Description
Method	Mandatory	See [RFC2616]	POST
Request-URI	Mandatory	See [RFC2616]	DCD Server address, per the applicable connection profile for the DCD interface
Host request-header	Mandatory	See [RFC2616]	the hostname or IP address of the DCD Server, per the applicable connection profile for the DCD interface
User-Agent request-header	Mandatory	See [RFC2616]	set to identify the host device (e.g. "vendor-model/version"), and the name and version of the DCD Client as user agent initiating the request
Accept request-header	Mandatory	See [RFC2616]	MIME media type of supported responses, i.e. "application/vnd.oma.dcd, multipart/related, multipart/mixed".
X-Wap-Profile	Mandatory	See [OMA-	URI of the User Agent Profile for the host device

request-header		UAPROF]	
Accept-Encoding request-header	Conditional	See [RFC2616]	Present if HTTP compression is supported. Set to the supported encodings, i.e. deflate and/or gzip
Content-Length entity-header	Mandatory	See [RFC2616]	set to the length of the entity-body
Content-Type entity-header	Mandatory	See [RFC2616]	MIME media type of the entity-body, one of "application/vnd.oma.dcd", "multipart/related", or "multipart/mixed", as applicable
Message-body	Mandatory	See Section 9	DCD-1 or DCD-3 message(s)

Table 167 Content of initial HTTP POST requests

Information Element	Req	Type	Description
Status-Line	Mandatory	See [RFC2616]	200 OK or HTTP error code upon HTTP Layer error
ETag entity-header	Mandatory	See [RFC2616]	A response header including an entity tag value uniquely identifying (in the scope of this DCD Server) the content file variant included in the message body
Content-Encoding entity-header	Mandatory	See [RFC2616]	set to the type of HTTP compression applied, if any
Content-Length entity-header	Mandatory	See [RFC2616]	set to the length of the entity-body
Content-Type entity-header	Mandatory	See [RFC2616]	MIME media type of the entity-body, one of "application/vnd.oma.dcd", "multipart/related", or "multipart/mixed", as applicable
Message-body	Mandatory	See Section 9	DCD-1, DCD-2, or DCD-3 message(s)

Table 168 Content of the response to the initial HTTP POST request, with an included DCD message

Information Element	Req	Type	Description
Method	Mandatory	See [RFC2616]	GET
Request-URI	Mandatory	See [RFC2616]	The URI pointing to the content file to be downloaded.
If-Match	Mandatory	See [RFC2616]	A request header, which SHALL include the ETag value from the initial download response.
Range	Mandatory	See [RFC2616]	A request header defining the byte range of the missing part of the content file.

Table 169 Content of HTTP Partial GET requests

Information Element	Req	Type	Description

Status-Line	Mandatory	See [RFC2616]	209 Partial Content or HTTP error code upon error
ETag	Mandatory	See [RFC2616]	A response header including an entity tag value uniquely identifying (in the scope of this DCD Server) the content file variant included in the message body
Content-Range	Mandatory	See [RFC2616]	An entity header specifying the byte range of the requested file, which is contained in the message body.
Message-body	Mandatory	Content	The requested byte range of the content file

Table 170 Content of the response message to HTTP Partial GET requests

13.Error Handling

If a transaction is not successfully completed, an error notification is sent to the requesting entity.

Only the first detected error is indicated in the error notification, and additional errors are not reflected in the error notification.

13.1 Error Conditions

The tables in the following sections summarize the interpretation and handling of error conditions based upon information at the transport and DCD layers. Normative requirements describing the expected DCD entity behaviour follow the tables.

The table columns represent:

- Error condition: the type of error that has been detected
- Detected by: the DCD entity detecting the error (DS = DCD Server, DC = DCD Client, CP = Content Provider, DECA = DCD Enabled Client Application)
- Response: the proper response to the error. If more than one entity response is shown in the table, the first one is the expected reaction of the error-detecting entity, and the second is the expected reaction of another entity when it receives notice of the error from the error-detecting entity. Examples of responses:
    - Fail: terminate handling of the message, and respond to the other DCD entity with the ErrorNotification message, indicating that message handling failed (server-level "fail") as a result of the error.
    - Retry: the action should be retried as indicated
    - Discard: the errored message/element should be discarded
- Description: scenario in which the error may be detected

DCD Servers and DCD Clients SHALL send the ErrorNotification message as the last message in the transaction that resulted in the error.

The overall approach to providing error notifications as described in the tables below, is for the error-detecting entity to select the error code, and other entities to simply to pass the error code on if applicable, in forwarded error notifications. As described in the tables below, the "fail" responses include an abbreviated description of the error notification parameters that should be provided, using this example as a template: ("error source", "error code", "error description"). The error-detecting entity identifies the source of the error, which will either be itself, or an interfacing entity for forwarded error notifications.

13.1.1 Generic Errors

This section addresses errors that may occur in a variety of situations.

Error Condition	Detected by	Response	Description
Security level error	DS	DS: fail ("server", "security level error") DC: retry with correct security level	Message with incorrect security level (e.g. HTTP request instead of HTTPS)
Malformed message	DS, DC	Discard message, log error	Message fails XML parser
Invalid parameter	DS	DS: fail ("server", "invalid parameter") DC: retry with correct parameter	Message with invalid parameter
	DC	DC: fail ("client", "invalid parameter") DS: retry with correct parameter	Message with invalid parameter

Error Condition	Detected by	Response	Description
Session ID error	DS	DS: fail (“server”, “invalid session ID”) DC: reactivate session	Message with invalid session ID
	DC	DC: reactivate session	Message with invalid session ID
Authentication error	DS	DS: fail (“server”, “authentication error”) DC: retry activation with correct authentication DC: provide an error response to affected application	Request with invalid authentication, e.g. during session activation, or an invalid authentication token received in any subsequent message.
	DC	DC: fail (“client”, “authentication error”) DS: retry activation with correct authentication, provide an error response to affected CP if any	Request with invalid authentication, e.g. during session activation, or an invalid authentication token received in any subsequent message.
Temporary failure	DS	DS: fail (“server”, “temporary failure”) DC: retry request after delay	Any “temporary failure” condition local to the DS
Request not allowed	DS	DS: fail (“server”, “not allowed”, reason) DC: fail request	Generic error for denied requests, e.g. per channel/service policy (channel subscription, personalization, on-demand channel update, suspend, resume, etc)
	DC	DC: fail(“application”, “not allowed”, reason) DS: fail request	Reception of an error notification from the DCD-Enabled Client Application, e.g. per channel/service policy (channel subscription validation, personalization, on-demand channel update, suspend, resume, etc)
Unregistered application	DC	DC: fail (“client”, “unregistered application”) DS: remove or update registration	Reception of a point-to-point message related to an unregistered DCD-Enabled Client Application.
Unsupported operation	DC	DC: fail (“client”, “unsupported operation”) DS: fail request or notification	Reception of an unsupported operation from the DCD Server, especially it is an optionally implemented interface in DCD Client side. (e.g. RequestForClientActivation, ClientDeactivationNotification, ChannelDiscoveryNotification, ContextualInforationUploadRequest)

Table 171 Generic Errors

Upon reception of a message delivered with an insufficient transport security level (e.g. HTTP instead of HTTPS), the DCD Server SHALL provide a failure response with error-code “security level error”.

Upon a failure response with error-code “security level error”, the DCD Client SHALL retry the request with a higher transport security level.

DCD Server and the DCD Client SHALL validate that the content item is packaged as defined in Section 9.1 and send an error message with error code “malformed message” if content package is malformed.

Upon reception of a malformed message (e.g. one that fails basic message parsing), the DCD Server or DCD Client SHALL discard the message, and take no further action.

Upon reception of a message with an invalid parameter, the DCD Server or DCD Client SHALL provide a failure response with error-code “invalid parameter”.

Upon a failure response with error-code “invalid parameter”, the DCD Client SHALL provide the correct parameter.

Upon reception of a message with an invalid session ID, the DCD Server SHALL provide a failure response with error-code "invalid session ID".

Upon a failure response with error-code "invalid session ID", the DCD Client SHALL establish a new session with the DCD Server.

Upon reception of a message with invalid session ID, the DCD Client SHALL discard the message, and establish a new session with the DCD Server.

Upon reception of a message that cannot be authenticated, the DCD Server SHALL provide a failure response with error-code "authentication error".

Upon a failure response with error-code "authentication error", the DCD Client SHALL retry the request with correct authentication. If necessary to provide correct authentication, the DCD Client SHALL establish a new session with the DCD Server.

Upon a failure response with error-code "authentication error", the DCD Client SHALL provide an error response to affected DCD-Enabled Client Applications.

Upon reception of a message that cannot be authenticated, the DCD Client SHALL provide a failure response with error-code "authentication error".

Upon a failure response with error-code "authentication error", the DCD Server SHALL retry the request with correct authentication. If necessary to provide correct authentication, the DCD Server SHALL establish a new session with the DCD Client.

Upon a failure response with error-code "authentication error", the DCD Server SHALL provide an error response to affected Content Providers.

If a temporary failure occurs during handling of a message, the DCD Server SHALL provide a failure response with error-code "temporary failure".

Upon a failure response with error-code "temporary failure", the DCD Client SHALL retry the request after a delay.

If a request is not allowed for generic reasons or specific reasons not addressed by other requirements, the DCD Server SHALL provide a failure response with error-code "not allowed" and an error-description identifying the reason.

Upon a DCD Server failure response with error-code "not allowed", the DCD Client SHALL stop the request, and not automatically retry it. If the request was initiated by a DCD-Enabled Client Application, the DCD Client SHALL respond with an error notification to the DCD-Enabled Client Application.

If the DCD Client receives a point-to-point message related to an unregistered DCD-Enabled Client Application, it SHALL send a failure response with error-code "unregistered application".

Upon a failure response with error-code "unregistered application", the DCD Server SHOULD stop the transaction, and not automatically retry it, until the registration inconsistency is resolved (e.g. the user switches back to another device for which the registration is valid).

If the DCD Client receives an unsupported request or notification (e.g. RequestForClientActivation, ClientDeactivationNotification), it SHALL send a failure response with error-code "unsupported operation".

Upon a failure response with error-code "unsupported operation", the DCD Server SHALL NOT retry this type of request or notification for the duration of this session.

13.1.2 Transport Errors

This section addresses errors related to events at the transport layer of DCD interfaces.

Error Condition	Detected by	Response	Description
HTTP error	DS	DS: response per HTTP [RFC2616] DC: fail or retry per [RFC2616]	HTTP errors (e.g. HTTP 4xx, 5xx status codes)
	DC	DC: response per HTTP [RFC2616] DS: fail or retry per [RFC2616]	HTTP errors (e.g. HTTP 4xx, 5xx status codes)
	DS	Upon failure, if related to a dependent DCD-1 or DCD-3 transaction: DS: fail ("content provider", "HTTP error", HTTP status code) DC: fail request	Generic response for HTTP errors of the Content Provider detected by DCD Server (e.g. HTTP 4xx, 5xx status codes)
Response timeout	DS, DC	retry request after delay	No response received to a transaction between DCD Server and DCD Client or between DCD Server and Content Provider, for which a response is expected.
	DS	Upon failure, if related to a dependent DCD-1 or DCD-3 transaction: DS: fail ("content provider", "server timeout") DC: retry request after delay	Timeout on Content Provider response from DCD Client though the DCD Server to the Content Provider. (e.g. content update request)
Client unreachable	DS	DS: fail ("server", "client unreachable") CP: retry request after delay	No DCD Client response (following any DCD Server retries, if applicable) to a request initiated by a Content Provider and delivered through the DCD Server to the DCD Client. (e.g. subscription validation)
Unsupported protocol version	DS	DS: fail ("unsupported protocol version") DC: retry with correct protocol version	Session activation request with unsupported DCD protocol version
uplink unavailable	DC	DC: Retry when possible	Transport uplink is not available – device cannot connect to the data network (for example in GSM/ UMTS the device fail to perform GPRS attach with the SGSN) – In most cases it is relatively persistent status.

Table 172 Transport Errors

If an HTTP error occurs (e.g. HTTP 4xx, 5xx status code related conditions) in the handling of a DCD Client message, the DCD Server SHALL provide a response as defined in [RFC2616].

Upon a response with an HTTP error status code (e.g. HTTP 4xx, 5xx status codes), the DCD Client SHALL respond as defined in [RFC2616].

Upon response timeout, the DCD Server or DCD Client SHALL retry the request after a delay.

The DCD Client SHOULD minimize the number of subsequent retries after receiving an error with error code "response timeout" on the transaction with the DCD Server. The timeout between retries is implementation specific.

The DCD Server SHOULD minimize the number of subsequent retries after receiving an error with error code "response timeout" on the transaction with the DCD Client. The timeout between retries is implementation specific.

If while serving a DCD Client request, the DCD Server times out on a Content Provider response, the DCD Server SHALL provide a failure response to the DCD Client with error-source "content provider" and error-code "server timeout".

Upon a failure response with error-source "content provider" and error-code "server timeout", the DCD Client SHALL retry the request after a delay.

If while serving a Content Provider request, the DCD Server fails to receive a DCD Client response, the DCD Server SHALL (following any DCD Server retries, if applicable) provide a failure response to the Content Provider with error-source "server" and error-code "client unreachable".

If while serving a DCD Client request, the DCD Server receives an HTTP error response (e.g. HTTP 4xx, 5xx status code) from the Content Provider, the DCD Server SHALL provide a failure response to the DCD Client with error-source "content provider", error-code "HTTP error", and error-description per the received HTTP status code.

Upon a failure response with error-source "content provider" and error-code "HTTP error", the DCD Client SHALL stop the request, and not automatically retry it.

If a ClientActivationRequest is received with an unsupported DCD protocol version attribute value, the DCD Server SHALL provide a failure response with error-code "unsupported protocol version".

Upon a failure response with error-code "unsupported protocol version", the DCD Client SHOULD retry the request with a DCD version supported by the DCD Server.

Upon request failure due to uplink unavailable, the DCD Client SHALL retry the request when the uplink is available

13.1.3 Activation Errors

This section addresses errors occurring during session activation.

Error Condition	Detected by	Response	Description
Authentication required	DS	DS: fail ("server", "authentication required") DC: retry with authentication	Session activation request with no authentication

Table 173 Activation Errors

If a ClientActivationRequest is received without necessary authentication or an incorrect authentication method, the DCD Server SHALL provide a failure response with error-code "authentication required".

Upon a failure response with error-code "authentication required", the DCD Client SHOULD retry the ClientActivationRequest with the correct authentication method.

13.1.4 Channel Subscription Errors

This section addresses errors occurring during channel subscription.

Error Condition	Detected by	Response	Description
Unregistered channel	DS	DS: fail ("server", "unregistered channel") DC: forward notification to DECA	Subscription request for unregistered channel, and the DCD Server has no policy to notify to the Content Provider
Purchase Option Outdated	DS	DS: fail ("server", "purchase option outdated")	The DCD Server gets the subscription request including the purchase-option-id is outdated.

Table 174 Channel Subscription Errors

If the DCD Server receives the channel subscription request for the unregistered channel and it has no interface or policy to notify to the Content Provider, it returns error notification with error-code "unregistered channel"

If the DCD Server gets the subscription request e.g. ChannelSubscriptionRequest, ChannelSubscriptionNotificationResponse with an outdated purchase-option-id, the DCD Server SHALL provider a failure notification with error-code “out dated”.

13.1.5 Delivery Errors

This section addresses errors occurring during content delivery.

Error Condition	Detected by	Response	Description
Invalid channel	DS	DS: fail (“server”, “invalid channel”) DC: remove channel metadata, notify applicable DECA’s	Request for invalid channel, e.g. expired or unknown
Channel not subscribed	DS	DS: fail (“server”. “channel not subscribed”) DC: remove channel metadata, notify applicable DECA’s	Request for non-subscribed channel update, or content item related to non-subscribed channel
	DC	DC: fail (“client”, “channel not subscribed”) DS: remove or update subscription	Reception of content for a non-subscribed channel
Content unavailable	DS	DS: fail (“server”, “content unavailable”) DC: if on-demand request, notify DECA	Request for unavailable content, e.g. expired from DS storage
Unusable Content	DC	DC: fail (“application”, “unusable content”, content-type) DS: forward notification to CP	Reception of error notification from DCD-Enabled Client Application for content which can not be rendered, e.g. application codec problem or unauthorized content due to DRM failure.
Delivery interrupted	DC	DC: fail (“client”, “delivery interrupted”) DS: when possible, retry or wait for DC retry	Content delivery or Content Submission was interrupted for various reasons, e.g. suspension or transport error.
	DS	DS: fail (“server”, ”delivery interrupted”) DC: when possible, retry	Content delivery or Content Submission was interrupted for various reasons, e.g. suspension or transport error.

Table 175 Content Delivery Errors

If a request is received with an invalid channel ID, the DCD Server SHALL provide a failure response with error-code “invalid channel”.

Upon a failure response with error-code “invalid channel”, he DCD Client SHALL remove information related to the channel from its storage, and notify applicable DCD-Enabled Client Applications that the channel is no longer valid.

If a request is received with a valid, but unsubscribed channel ID, the DCD Server SHALL provide a failure response with error-code “unsubscribed channel”.

Upon a failure response with error-code “unsubscribed channel”, the DCD Client SHALL remove information related to the channel from its storage, and notify applicable DCD-Enabled Client Applications that the channel is no longer valid.

If the DCD Client receives content for an unsubscribed channel in a point-to-point transaction, it SHALL provide a failure response with error-code “unsubscribed channel”.

Upon a failure response with error-code “unsubscribed channel”, the DCD Server SHOULD resolve the channel subscription inconsistency, e.g. by removing the channel subscription, or updating the DCD Client with the correct subscription information.

If a request is received for an unavailable content item, the DCD Server SHALL provide a failure response with error-code "content unavailable".

Upon a failure response with error-code "content unavailable", the DCD Client SHALL fail the content item retrieval and not automatically retry it. If the request was initiated by a DCD-Enabled Client Application, the DCD Client SHALL provide a corresponding failure response.

Upon reception of a error notification with error code "unusable content" from the DCD Enabled Client Application, if the content item was not received over a broadcast bearer the DCD Client SHALL forward the error notification to the DCD Server. Upon reception of a error notification with error code "unusable content" from a DCD Client, the DCD Server SHALL pass the error notification to the Content Provider.

If content delivery or content submission is interrupted, e.g. upon channel suspension, the DCD Client SHALL provide a failure notification with error-code "delivery interrupted" and errored-parameter as the number of bytes received.

Upon a failure notification with error-code "delivery interrupted", the DCD Server SHOULD retry the transaction when allowed, e.g. when channel resumption occurs.

If content delivery or content submission is interrupted, e.g. upon channel suspension, the DCD Server SHALL provide a failure notification with error-code "delivery interrupted" and errored-parameter as the number of bytes delivered.

Upon a failure notification with error-code "delivery interrupted", the DCD Client SHOULD retry the transaction when allowed, e.g. when channel resumption occurs.

13.2 Error Messages

13.2.1 Error Notification in DCD-1 and DCD-3

The Error Notification transaction (see Figure 79: Error Notification Transaction) is used by the DCD Server or DCD Client to notify the other DCD entity about an error condition that has occurred .

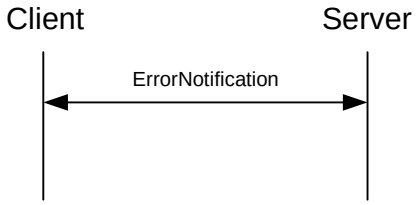


Figure 79: Error Notification Transaction

13.2.1.1 Messages and Information Elements

Message	Implementation	Direction
ErrorNotification	Mandatory	DCD Client ← DCD Server or DCD Client → DCD Server

Table 176 Message directions for ErrorNotification message

Information Element	Req	Type	Description
Session-ID	Conditional	String	Session identifier. The Session-ID is unique within the service provider domain.

			Included only if the error occurs after session is established.
Message-ID	Conditional	String	Identifies this message. The Message-ID is unique within a current session. Message identifier consists of the transaction identifier suffixed by two numeric characters for message index within the transaction. Transaction identifier offset is unique within the current session and identical for all messages within the transaction. If the error is occurred before the session activation, the Message-ID is empty.
Error-Severity	Mandatory	String	The impact of the error condition on the handling of the indicated message, with possible values “fail”, “warn”.
Error-Source	Mandatory	Enumerated	Source of the error. Possible values 0-3 (application, client, server, content provider)
Error-Code	Mandatory	Integer	Numeric code for the error that has occurred.
Error-Description	Conditional	String	Free text describing the error. Present if provided by error source. For forwarded error notifications the entity SHALL pass this parameter unmodified.
Errored-Parameter	Conditional	String	Name of the message element or attribute that was found to be in error. SHALL be present for error codes “invalid parameter”, “unusable content”, “delivery interrupted” , “HTTP error”.

Table 177 Information elements in ErrorNotification message

13.2.2 Error Notification in DCD-CPR, DCD-CPDE, DCD-CAR, DCD-CADE

The Error Notification transaction (see Figure 80: Error Notification Transaction between DCD Server and Content Provider) is used between DCD Servers and Content Providers, or DCD Clients and DCD Enabled Client Applications (see Figure 81: Error Notification Transaction between DCD Client and DCD Enabled Client Application), to notify the other entity about an error condition that has occurred .

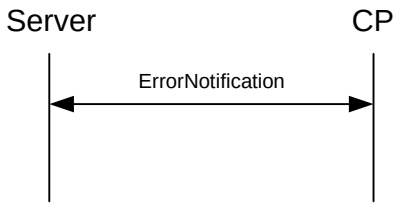


Figure 80: Error Notification Transaction between DCD Server and Content Provider

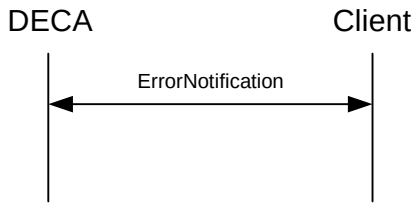


Figure 81: Error Notification Transaction between DCD Client and DCD Enabled Client Application

13.2.2.1 Messages and Information Elements

Message	Implementation	Direction
ErrorNotification	Mandatory	Content Provider ←DCD Server or Content Provider → DCD Server or DCD-Enabled Client Application → DCD Client or DCD Enabled Client Application ← DCD Client

Table 178 Message directions for ErrorNotification message

Information Element	Req	Type	Description
Error-Source	Mandatory	Enumerated	Source of the error. Possible values 0-3 (application, client, server, content provider)
Error-Code	Mandatory	Integer	Numeric code for the error that has occurred.
Error-Description	Optional	String	Free text describing the error. May contain information about the message that was being handled when the error occurred. For forwarded error notifications, the DCD Server and DCD Client SHALL pass this parameter unmodified.
Errored-Parameter	Conditional	String	Name of the message element or attribute that was found to be in error. SHALL be present for error codes “invalid parameter”, “unusable content”, “delivery interrupted”, HTTP error”.

Table 179 Information elements in ErrorNotification message

13.3 Error Codes

This section provides the numeric values and text names for the error codes.

Error Code	Error Description
001	security level error
002	invalid parameter
003	invalid session id
004	authentication error
005	temporary failure
006	not allowed
007	unregistered application

Error Code	Error Description
008	server timeout
009	client unreachable
010	unsupported protocol version
011	authentication required
012	content unavailable
013	unusable content
014	HTTP error
015	invalid channel
016	channel not subscribed
017	delivery interrupted
018	unregistered channel
019	purchase option outdated
020	uplink unavailable
021	unsupported operation

13.4 Channel Metadata Mapping to DCD Messages

This section provides a mapping between Channel Metadata subsets and DCD messages in which the subsets may occur. In the table below:

- "M" means the Channel Metadata subset SHALL be present in the message, if Channel Metadata is present according to Section 7
- "C" means the Channel Metadata subset SHALL be present in the message under the conditions specified for the message (see Section 7), or if a previous related message in a transaction contained the Channel Metadata it may conditionally be present if relevant to the associated message over the next interface
- "O" means the Channel Metadata subset MAY be present in the message

Channel Metadata item	DCD-CPR ChannelRegistrationRequest	DCD-CPR ChannelRegistrationResponse	DCD-CAR ApplicationRegistrationRequest	DCD-3 ApplicationRegistrationRequest	DCD-3 ApplicationRegistrationResponse	DCD-CAR ApplicationRegistrationResponse	DCD-CAR SubscriptionRequest	DCD-CAR SubscriptionUpdateRequest	DCD-3 ChannelSubscriptionRequest	DCD-CPR SubscriptionRequest	DCD-CPR SubscriptionResponse	DCD-3 ChannelSubscriptionResponse	DCD-CAR SubscriptionResponse	DCD-CPR SubscriptionNotification	DCD-3 ChannelSubscriptionNotification	DCD-CAR SubscriptionValidationRequest	DCD-CAR SubscriptionValidationResponse	DCD-3 ChannelSubscriptionNotificationResponse	DCD-CPR SubscriptionNotificationResponse	DCD-3 ChannelDiscoveryInfo	DCD-CADE ChannelDiscoveryInfo	DCD-CADE ChannelDiscoveryResponse	DCD-3 ChannelMetadataUpdate	DCD-CAR ChannelMetadataUpdate
channel-selection-metadata			M	M																				
delivery-personalization-metadata							M	M	M								M	M						
general-channel-metadata	O	M			M	M								M	M	M				M	M	M	M	M
charging-metadata	O	O			O	C								O	C	C				O	C	C	O	C
delivery-preferences-metadata	O	O			O						M	M		O	C					*			O	
channel-publication-metadata	M	O																						

Table note: (*) In case of broadcast, the dcd-2-broadcast-profile-name and dcd-2-broadcast-profile parts of the delivery-preferences-metadata subset could be delivered in the DCD-3 ChannelDiscoveryInfo.

13.5 Error Code Mapping to DCD Messages

This section provides a mapping between error codes and DCD interface messages. In the table below, "*" means the error code may result from processing of the message in typical error cases (i.e. not as a result of data corruption or software error), and as a result be conveyed via the ErrorNotification message. Unshaded cells are the first message in transactions, or subsequent messages where a response is typically expected or may be expected due to message parameters e.g. confirmation request. Shaded cells are the last message in a transaction.

Message	Direction	invalid parameter	invalid session id	authentication error	temporary failure	not allowed	unregistered application	server timeout	client unreachable	unsupported protocol version	authentication required	content unavailable	unusable content	HTTP error	invalid channel	channel not subscribed	delivery interrupted	unregistered channel	purchase option outdated	unsupported operation
DCD-1 ContentUpdateRequest	DC → DS	*	*	*	*			*				*		*		*	*			
DCD-1 ContentUpdateResponse	DC ← DS	*	*										*	*						
DCD-1 ContentDeliveryConfirmation	DC → DS	*	*											*						
DCD-1 ContentSubmitRequest	DC → DS	*	*	*	*	*		*						*		*	*			
DCD-1 ContentSubmitConfirmation	DC ← DS	*	*											*						
DCD-2 ContentUpdatePush	DC ← DS	*	*																	
DCD-2 ContentUpdateNotification	DC ← DS	*	*																	
DCD-3 ClientActivationRequest	DC → DS	*		*	*	*				*	*			*						
DCD-3 ClientActivationResponse	DC ← DS	*	*	*										*						
DCD-3 RequestForClientActivation	DC ← DS																			*
DCD-3 ClientDeactivationRequest	DC → DS	*	*	*	*	*								*						
DCD-3 ClientDeactivationResponse	DC ← DS	*	*											*						
DCD-3 ClientDeactivationNotification	DC ← DS	*	*																	*
DCD-3 ApplicationRegistrationRequest	DC → DS	*	*	*	*	*								*						
DCD-3 ApplicationRegistrationResponse	DC ← DS	*	*											*						
DCD-3 ApplicationDeregistrationNotification	DC → DS	*	*	*	*	*	*							*						
DCD-3 ApplicationDeregistrationConfirmation	DC ← DS	*	*											*						
DCD-3 ContextualInformationUploadRequest	DC ← DS	*	*											*						*
DCD-3 ContextualInformationUpload	DC → DS	*	*	*										*						
DCD-3 RequestForUsageTrackingReport	DC ← DS	*	*				*													
DCD-3 UsageTrackingReport	DC → DS	*	*	*			*							*						

DCD-3 ChannelSubscriptionRequest	DC → DS	*	*	*	*	*	*	*						*	*			*	*	*
DCD-3 ChannelSubscriptionResponse	DC ← DS	*	*											*						
DCD-3 ChannelUnsubscriptionRequest	DC → DS	*	*	*	*	*	*	*						*						
DCD-3 ChannelUnsubscriptionResponse	DC ← DS	*	*											*						
DCD-3 ChannelUnsubscriptionNotification	DC ← DS	*	*				*													
DCD-3 ChannelUnsubscriptionConfirmation	DC → DS	*	*	*										*						
DCD-3 ChannelSubscriptionNotification	DC ← DS	*	*			*	*													
DCD-3 ChannelSubscriptionNotificationResponse	DC → DS	*	*	*										*					*	
DCD-3 ChannelDiscoveryInfo	DC ← DS	*	*				*							*						
DCD-3 ChannelDiscoveryConfirmation	DC → DS	*	*	*										*						
DCD-3 ChannelDiscoveryNotification	DC ← DS	*	*				*													*
DCD-3 ChannelDiscoveryRequest	DC → DS	*	*	*	*	*	*							*						
DCD-3 ChannelSuspendRequest	DC → DS	*	*	*	*	*								*		*				
DCD-3 ChannelSuspendResponse	DC ← DS	*	*											*						
DCD-3 ChannelResumeRequest	DC → DS	*	*	*	*	*								*		*				
DCD-3 ChannelResumeResponse	DC ← DS	*	*											*						
DCD-3 ChannelSuspendNotification	DC ← DS	*	*																	
DCD-3 ChannelResumeNotification	DC ← DS	*	*																	
DCD-3 ConnectionProfileConfirmation	DC → DS	*	*											*						
DCD-3 ConnectionProfileUpdate	DC ← DS	*	*											*						
DCD-3 ContentRepairRequest	DC → DS	*	*	*	*	*		*				*	*	*		*	*			
DCD-3 ContentRepairResponse	DC ← DS	*	*											*						
DCD-3 ChannelMetadataUpdate	DC ← DS	*	*													*				
DCD-3 ChannelMetadataUpdateConfirmation	DC → DS	*	*	*										*						
DCD-CPR SubscriptionRequest	DS → CP	*		*	*	*								*						
DCD-CPR SubscriptionResponse	DS ← CP	*												*						
DCD-CPR UnsubscriptionNotification	DS → CP	*		*										*						
DCD-CPR SubscriptionNotification	DS ← CP	*			*	*			*					*						
DCD-CPR SubscriptionNotificationResponse	DS → CP	*												*						
DCD-CPR SubscriptionUpdate	DS ← CP	*												*						
DCD-CPR RequestForChannelRegistration	DS → CP	*			*	*								*						
DCD-CPR ChannelRegistrationRequest	DS ← CP	*			*	*								*						
DCD-CPR ChannelRegistrationResponse	DS → CP	*												*						
DCD-CPR ChannelDeregistrationNotification	DS → CP	*			*									*						
DCD-CPR ChannelDeregistrationConfirmation	DS ← CP	*												*						
DCD-CPR ChannelDeregistrationRequest	DS ← CP	*			*									*						
DCD-CPR ChannelDeregistrationResponse	DS → CP	*												*						
DCD-CPR RequestForUsageReport	DS ← CP	*			*									*						

DCD-CPR UsageReport	DS → CP	*											*						
DCD-CPDE ChannelSuspendRequest	DS → CP	*			*								*						
DCD-CPDE ChannelSuspendResponse	DS ← CP	*											*						
DCD-CPDE ChannelResumeRequest	DS → CP	*			*								*						
DCD-CPDE ChannelResumeResponse	DS ← CP	*											*						
DCD-CPDE ChannelSuspendNotification	DS ← CP	*											*						
DCD-CPDE ChannelResumeNotification	DS ← CP	*			*	*							*						
DCD-CPDE ChannelResumeConfirmation	DS → CP	*											*						
DCD-CPDE ContentUpdateRequest	DS → CP	*			*								*						
DCD-CPDE ContentUpdateResponse	DS ← CP	*											*						
DCD-CPDE ContentUpdate	DS ← CP	*			*								*						
DCD-CAR ApplicationRegistrationRequest	DECA → DC	*			*	*							*						
DCD-CAR ApplicationRegistrationResponse	DECA ← DC	*																	
DCD-CAR ApplicationDeregistrationRequest	DECA → DC	*			*		*	*					*						
DCD-CAR ApplicationDeregistrationResponse	DECA ← DC	*																	
DCD-CAR SubscriptionRequest	DECA → DC	*			*	*	*	*						*	*			*	*	*
DCD-CAR SubscriptionResponse	DECA ← DC	*																	
DCD-CAR SubscriptionUpdateRequest	DECA → DC	*			*	*	*	*						*		*				
DCD-CAR SubscriptionUpdateResponse	DECA ← DC	*																	
DCD-CAR UnsubscriptionRequest	DECA → DC	*			*	*	*	*						*		*				
DCD-CAR UnsubscriptionResponse	DECA ← DC	*																	
DCD-CAR UnsubscriptionNotification	DECA ← DC	*																	
DCD-CAR SubscriptionValidationRequest	DECA ← DC	*				*													
DCD-CAR SubscriptionValidationResponse	DECA → DC	*																*	
DCD-CAR ChannelMetadataUpdate	DECA ← DC	*																	
DCD-CADE ChannelDiscoveryRequest	DECA → DC	*			*	*	*						*						
DCD-CADE ChannelDiscoveryResponse	DECA ← DC	*																	
DCD-CADE ContentRequest	DECA → DC	*			*	*	*	*			*		*		*	*			
DCD-CADE Content	DECA ← DC	*										*							
DCD-CADE ContentSubmitConfirmation	DECA ← DC	*											*						
DCD-CADE ContentSubmitRequest	DECA → DC	*			*	*	*	*					*		*	*			
DCD-CADE ChannelSuspendRequest	DECA → DC	*			*	*	*						*		*				
DCD-CADE ChannelSuspendResponse	DECA ← DC	*																	
DCD-CADE ChannelResumeRequest	DECA → DC	*			*	*	*						*		*				
DCD-CADE ChannelResumeResponse	DECA ← DC	*																	
DCD-CADE ChannelSuspendNotification	DECA ← DC	*																	
DCD-CADE ChannelResumeNotification	DECA ← DC	*																	
DCD-CADE ChannelDiscoveryInfo	DECA ← DC	*																	

14.Terminal Capability Disclosure

DCD Clients and DCD Servers SHALL support the mechanisms specified in [OMA-UAPROF] for disclosure of DCD-related device characteristics.

14.1 UAProf Schema Extension

During the DCD Client Activation process, the DCD Client SHALL disclose its UAProf including attributes from the DCD Characteristics component defined below.

DCD Clients MAY use attributes from other components of the UAProf schema to indicate capabilities of the other device components.

14.1.1 DCD Attributes in Other Components of the UAProf Schema

The UAProf specification includes a schema containing attributes that describe the client hardware, the browser user-agent, network characteristics and more. Some of the attributes included in the aforementioned specification also apply to the DCD Client, e.g. “CpuType”, and “PushMessageSize”. For a complete reference to the attributes available in the UAProf schema, please see [OMA-UAPROF].

14.1.2 Summary of the DCD Characteristics Component

A normative description can be found in Appendix D. The table below summarizes the attributes defined within the DCD Characteristics component.

Component: DcdCharacteristics				
Attribute	**Description**	**Resolution**	**Datatype**	**Example(s)**
DcdClientName	Name of the DCD Client, in vendor / client / version format.	Locked	Literal	"SPARQY/DC-Leet/1.0 "
DcdVersion	DCD versions supported by the DCD Client, conveyed as majorVersionNumber.minorVersionNumber.	Locked	Literal bag	“1.0”, “1.1”
DcdSupportedBearers	Transport bearers for DCD that are supported by the device.	Locked	Literal bag	“EDGE”, “UMTS”, “CDMA2000”, “GAN”, “WiMAX”, “LTE”, “802.11”, “BCAST”, “CBS”
DcdSupportedProtocols	Transport protocols for DCD that are supported by the device for P2P bearers.	Locked	Literal bag	“HTTP”, “HTTPS”
DcdSupportedMethods	Content delivery methods supported by the DCD Client.	Locked	Literal bag	“Pull”, “Push”
DcdAuthMethods	DCD session authentication methods supported by the device.	Locked	Literal bag	“digest-user”, “digest-gba”, “x509”
DcdAcceptSchema	DCD packaging schema supported by the device.	Locked	Literal bag	“dcd-xml”, “atom+dcd-xml”, “rss+dcd-xml”

Note: The DCDSupportedBearers values shown are specifically defined in the DCD-XML schema for use in DCD, and supported by the DCD-XML compact binary encoding. Additional values that may apply for different networks are defined by the Open Mobile Naming Authority (OMNA) and described at the OMA website at URL: http://www.openmobilealliance.org/tech/profiles/ccppschema-20070511.aspx.

15. DCD Compact Encoding

15.1 Introduction

WBXML 1.3 [WBXML] is a simple method that allows compacting XML documents in a lossless manner. A WBXML decoder processes a WBXML encoded document by interpreting it byte-by-byte. Some bytes represent decoding instructions, some represent XML element start tags, attribute names or attribute values. The decoding process is stateful. The decoder maintains one global state, which determines whether it is processing elements, or attributes. Within each state, the decoder maintains an independent notion of a selected code page.

15.1.1 Associating XML Documents with WBXML Token Values

An external typing system has to be used to associate XML documents with WBXML token values.

If the document is transported by WSP, HTTP, CBS, the media type has to be used. Since the token values are associated with the media type, and not a particular version of the document type definition, the tokeniser is independent of the document type version; and can tokenise any version of the document type. To ensure compatibility between different versions of user-agents and tokenisers, the user-agent has to support both the binary token value and the literal value for all tags, attribute names, and attribute values.

15.1.2 WBXML Document Format

A WBXML document construct as a sequence of the following elements:

1. WBXML version used.
2. The Document Public Identifier
3. Character set used to encode strings
4. A string table.
5. WBXML Body containing the encoded XML body

15.2 DCD implementation

DCD enabler SHALL use the compact encoding only for downlink (i.e. DCD Server to DCD Client) when using the WAP-Push bearer and/or the CBS bearer.

15.2.1 DCD Media Type

The following Media Types are supported by DCD and are carried by the relevant mechanism depending on the transport bearer that is used i.e. HTTP headers, WSP or DCD Header for CBS (see [DCD-TS-CBS]).

Textual form (XML): application/vnd.oma.dcd

Tokenised form (WBXML): application/vnd.oma.dcdc

When used over WAP-PUSH or over CBS the same tokens values are assigned:

Media-Type	WSP Assigned number	CBS Assigned number
application/vnd.oma.dcdc	0x54	0x54
application/vnd.oma.dcd	0x53	0x53

Table 180 Media Type tokens assignment

15.2.2 WBXML version

For this specification, WBXML 1.3 is used. This is represented by the version byte value 0x03 as specified in [WBXML].

15.2.3 WBXML Document Public Identifier

The Document Public Identifier is "-//OMA//DCD 1.0//EN" is registered by OMNA, and is represented as a single byte value 0x15.

15.2.4 Character set

The character set to be used is UTF-8, represented as one byte value 0x6a.
The value of the WBXML Charset field is the MIBEnum value assigned by the IANA for the character encoding ((see [IANACharset])).

15.2.5 String table

The string table is always empty for DCD, this is represented as one byte values 0x00 (representing a string table length of 0).

15.2.6 WBXML Body

Following the above header's parameters, there is a body. This is a sequence of codes from the code pages, signaling processing instructions, element tags, attribute names and values. The decoder starts with tag code page 0, and attribute code page 0.

The next sections contain tables of the DCD assigned numbers. OMA is responsible for administering the values. The following describes the process requirements for administration of DCD assigned numbers:

- New entities SHALL be added at the end of any tables.
- When removing an entity in one of those tables, the assigned number SHALL be deprecated and it SHALL NOT be re-used for another entity.
- If the encoding rules of an entity need to be changed, a new entity SHALL be created.

15.2.6.1 Element tag Tokens

The following token codes represent tags in code page zero (0). All numbers are in hexadecimal.

Tag Name	Token [0x]
Adaptation-Capability	5
Application-ID	6
Available-Bearers-Report	7
Broadcast-Service-ID	8
ChannelDiscoveryInfo_overDCD3	9
ChannelDiscoveryNotification	A
Channel-Icon	B
Channel-ID	C
Channel-ID-Filter	D

Channel-IDs	E
Channel-Metadata	F
ChannelMetadataUpdate _overDCD3	10
ChannelResumeNotification _overDCD3	11
Channels-Added	12
Channels-Removed	13
ChannelSubscriptionNotification _overDCD3	14
Channels-Updated	15
ChannelSuspendNotification_overDCD3	16
ChannelUnsubscriptionNotification_overDCD3	17
Charging-Metadata-Type	18
Charging-Rules	19
ClientDeactivationNotification	1A
Client-Package	1B
Content	1C
Content-ID-Filter	1D
Content-Type-Filter	1E
ContentUpdateNotification _overDCD2	1F
ContentUpdatePush _overDCD2	20
ContextualInformationUploadRequest	21
DC-CNT-Metadata	22
DCD-Content-Reference	23
Free-Storage-Report	24
Message-ID	25
Mime-Type-Filter	26
Policy-ID	27
Preferred-Bearers-Report	28
Price	29
Purchase-Options	2A
Reason	2B
Report-Frequency	2C

Reporting-Schedule	2D
Report-Policy_CI	2E
Report-Policy_UT	2F
RequestForClientActivation	30
RequestForUsageTrackingReport	31
Reserved-Storage-Report	32
Roaming-Status-Report	33
Session-ID	34
Storage-Report-Policy	35
Subscription-ID	36
Tracking-Schedule	37
Usage-Report-Server-Address	38

Table 181 Element Tag Tokens

15.2.6.2 Attribute Start Tokens

The following token codes represent the start of an attribute in code page zero (0). All numbers are in hexadecimal.

Attribute Name	Attribute Value Prefix	Token [0x]
COMMON ATTRIBUTES		
xsi:schemaLocation		5
xsi:schemaLocation	"http://www.openmobilealliance.com/oma-dcd/1.0"	6
xmlns:dcd		7
xmlns:dcd	"http://www.openmobilealliance.com/oma-dcd/1.0"	8
xmlns:xsi	"http://www.w3.org/2001/XMLSchema-instance"	9
Amount		A
Aux-Content-Link		B
Channel-Description		C
Channel-ID		D
Channel-Name		E
Content-Address		F
Content-Delivery-Notification		10
Content-Encoding		11
Content-Expiration		12
Content-ID		13
Content-Length		14
Content-Name		15
Content-Price		16
Content-Protection		17
Content-Storage-Location		18
Content-Types		19
Content-Updated		1A
Cost-Information		1B
Currency		1C
Delivery-Priority	1 - Low	1D
Delivery-Priority	2 - Medium	1E
Delivery-Priority	3 - High	1F

Attribute Name	Attribute Value Prefix	Token [0x]
Emergency-Content		20
Genre		21
Matching-Applications		22
Mime-Type		23
Mime-Types		24
Parental-Rating		25
Purchase-Option-Id		26
Replaces-Content-ID		27
Storage-Reservation		28
subscription-required		29
Updated		2A

Table 182 Common Attribute Start Tokens

15.2.6.3 Attribute Value Tokens

The following token codes represent attribute values in code page zero (0). All numbers are in hexadecimal.

Attribute Value	Token [0x]
.com/	85
.edu/	86
.net/	87
.org/	88
.mobi/	89
www.	8A
.com	8B
.co	8C
wap	8D
http://	8E
http://www.	8F
https://	90
https://www.	91

Attribute Value	Token [0x]
False	92
True	93
EUR	94
USD	95
GBP	96
802.11	97
any-change	98
BCAST	99
CBS	9A
exhausted	9B
home	9C
international	9D
LTE	9E
on-demand	9F
roam	A0
UMTS	A1
WiMAX	A2

Table 183 Attribute Value Tokens

15.2.7 Encoding of data type DateTime

DateTime data MUST be encoded as OPAQUE data with each number in the string represented by its 4-bit binary value. Any non-numerical characters (“T”, “Z”, “-“, and “:”) are discarded. Trailing zeros (from right to left) MUST be pair-wise omitted.

For example, “2008-07-21T06:40:00Z” is encoded into six octets as follows:

Number	“2”	“0”	“0”	“8”	“0”	“7”	“2”	“1”	“0”	“6”	“4”	“0”	“0”	“0”
Binary value	0010	0000	0000	1000	0000	0111	0010	0001	0000	0110	0100	0000	0000	0000
Octet (hex)	00100000 (20)		00001000 (08)		00000111 (07)		00100001 (21)		00000110 (06)		01000000 (40)		omitted	

16. Connection Profile Management

16.1 Default Connection Profile Management by DCD MO

[DCD-MO-TS] describes the OMA Dynamic Content Delivery (DCD) Management Object (MO) syntax that that enables management of default DCD connection profiles. DCD MO's are created and managed by the OMA Device Management (DM) entities DM Server and DM Client. The normative requirements on the OMA DM entities are addressed by the OMA DM enabler. Those requirements include the generic handling of management objects. This section describes the normative requirements on DCD enabler entities, specifically DCD Clients, related to DCD connection profiles as managed through OMA DM.

Note that device-internal mechanisms for establishment of a DCD MO are unspecified, e.g. population of the DCD MO from default DCD Client configuration data. Also unspecified are the internal interfaces between DM Clients and DCD Clients, through which the DCD Client is notified of changes to the DCD MO. The normative requirements on the DCD Client address how DCD MO changes, either a new MO or changes to an already-established MO, are incorporated by the DCD Client into its internal connection profile configuration.

16.2 Handling of Connection Profile Changes

When a DCD Client becomes aware of changes to connection profiles managed as DCD MO's through OMA DM or via the DCD-3 Connection Profile Update transaction:

- For the DCD-3 Connection Profile Update transaction, within the scope of the connection profiles related to each DCD interface (DCD-1, DCD-2, and DCD-3), the DCD Client SHALL find the applicable connection profile instances to be updated by matching the "Name" attribute. If there is no "Name" attribute, the default profile is updated.
- For changed connection profiles, DCD Clients SHALL add/update/remove the attributes and elements as indicated.
- For new connection profiles, DCD Clients SHALL use the attributes and elements as indicated.
- For changes to DCD-3 connection profiles related to an existing DCD session which the Session-TTL value is zero, the DCD Client SHALL apply the changes immediately, i.e. establish a new DCD session per the connection profile. If the Session-TTL value is non-zero, the DCD Client SHALL apply the changes for the next DCD session.
- For changes to DCD-1 and DCD-2 connection profiles related to an existing DCD session, the DCD Client SHALL apply the changes for subsequent transactions.
- For removed connection profiles, DCD Clients SHALL apply the changes immediately.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-DCD_Semantics-V1_0-20110705-A	05 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0224-INP_DCD_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 DCD Client

Item	Function	Reference	Status	Requirement
DCD-C-001-M	Client Activation and Deactivation	5.1, 5.4, 6.1.1	M	
DCD-C-002-M	Application Registration and Deregistration	5.2.1, 5.3.1, 6.1.2	M	
DCD-C-003-M	Internal Subscription and Unsubscription	5.5, 6.1.3.1, 6.1.3.4	M	
DCD-C-004-O	External Subscription and Unsubscription	5.5, 6.1.3.2, 6.1.3.4	O	
DCD-C-005-O	Subscription Personalization	5.5.2, 6.1.3.3	O	
DCD-C-006-O	Channel Metadata Update	6.1.4	O	
DCD-C-007-M	Content Update Request	5.6, 6.1.5.2, 6.1.5.3, 6.1.5.6	M	
DCD-C-008-M	Content Update Push	5.6, 6.1.5.4, 6.1.5.6	M	
DCD-C-009-M	Content Update Notification	5.6, 6.1.5.5, 6.1.5.6	M	
DCD-C-010-M	Content Item Handling	6.1.5.6, 8.3	M	
DCD-C-011-O	DCD-Managed Content Storage	6.1.5.6	O	
DCD-C-012-M	Content Submission	5.7, 6.1.6	M	
DCD-C-013-O	DECA-Initiated Suspension and Resumption	5.8, 6.1.7.1, 6.1.7.2	O	
DCD-C-014-M	DCD Client-Initiated Suspension and Resumption	5.8, 6.1.7.1, 6.1.7.2	M	
DCD-C-015-O	DCD Server-Initiated Suspension and Resumption	5.8, 6.1.7.1, 6.1.7.3	O	
DCD-C-016-O	DECA-Initiated Channel Discovery	5.2.2, 6.1.8.1, 6.1.8.3	O	
DCD-C-017-M	DCD Server-Initiated Channel Discovery	5.2.2, 6.1.8.1, 6.1.8.2, 6.1.8.3, 6.1.8.4,	M	
DCD-C-018-O	Usage Tracking Report	6.1.9	O	
DCD-C-019-O	Contextual Information Upload	6.1.10	O	
DCD-C-020-M	Charging	6.1.11	M	
DCD-C-021-O	Content Repair	6.1.12	O	

B.2 SCR for DCD Server

Item	Function	Reference	Status	Requirement
DCD-S-001-M	Client Activation and Deactivation	5.1, 5.4, 6.2.1	M	
DCD-S-002-M	Application Registration and Deregistration	5.2.1, 5.3.1	M	
DCD-S-003-M	Channel Registration and Deregistration	5.2.2, 5.3.2, 6.2.3	M	
DCD-S-004-M	Internal Subscription and Unsubscription	5.5, 6.2.4.1, 6.2.4.5	M	
DCD-S-005-O	External Subscription and Unsubscription	5.5, 5.5.3, 6.2.4.2, 6.2.4.5	O	
DCD-S-006-O	Subscription Compatibility Verification	6.2.4.3	O	
DCD-S-007-O	Subscription Update	6.2.4.4	O	
DCD-S-008-O	Subscription Personalization	5.5.2	O	
DCD-S-009-O	Channel Metadata Update	6.2.5	O	
DCD-S-010-O	Content Publication	6.2.6.2	O	
DCD-S-011-M	Content Update Request	5.6, 6.2.6.3, 6.2.6.4, 6.2.6.5, 6.2.6.6	M	
DCD-S-012-M	Content Update Push	5.6, 6.2.6.7	M	
DCD-S-013-M	Content Update Notification	5.6, 6.2.6.8	M	
DCD-S-014-M	Content Item Handling	6.2.6.9, 6.2.6.10, 8.3	M	
DCD-S-015-M	Content Submission	5.7, 6.2.7	M	
DCD-S-016-M	DCD Client-Initiated Suspension and Resumption	5.8, 6.2.8.2	M	
DCD-S-017-O	DCD Server-Initiated Suspension and Resumption	5.8, 6.2.8.3	O	
DCD-S-018-O	Content Provider-Initiated Suspension and Resumption	6.2.8.4	O	
DCD-S-019-O	DCD Client-Initiated Channel Discovery	5.2.2, 6.2.9.2	O	
DCD-S-020-M	DCD Server-Initiated Channel Discovery	5.2.2, 6.2.9.3, 6.2.9.4	M	
DCD-S-021-O	Channel Guide Compatibility Verification	6.2.9.1	O	
DCD-S-022-O	Usage Tracking Report	6.2.10	O	
DCD-S-023-O	Contextual Information Upload	6.2.11	O	
DCD-S-024-M	Charging	6.2.12	M	

Item	Function	Reference	Status	Requirement
DCD-S-025-O	Content Repair	6.2.13	O	

Appendix C. Content Delivery message transformations in the DCD enabler

This appendix shows examples of Content Delivery messages as sent/received by a DCD Server and a DCD Client. For simplicity, only the ContentUpdate (over DCD-CADE) → ContentUpdatePush (over DCD-2) → Content (over DCD-CADE) message chain is demonstrated in the examples below. Other Content Delivery messages (e.g. ContentUpdateResponse) are transformed in a similar manner. The examples shown are for dcd-xml and atom+dcd-xml packaging schemas (see Section 9).

C.1 dcd-xml Packaging Schema

The XML fragment below shows an example of a well-formed ContentUpdate message over DCD-CPDE interface (see Section 7.2.2.3). When the DCD Server receives this message, it extracts the server-related content metadata associated with the content package and uses this metadata to process the client envelope. The DCD Sever may add or update DCD Client related content metadata parameters, if necessary.

```
<dcd>
    <ContentUpdate_overCPDE>
        <server-package xmlns:dcd="http://www.openmobilealliance.com/oma-dcd/1.0">
                <ds-cnt-metadata content-block-id="0xA017" content-id="0xB12F" deliver-at="2009-10-10T18:30:02Z" deliver-when-
                roaming=true network-preferences="WiMAX;UMTS;LTE" parental-rating="G" delivery-priority="1 - Low" deliver-per-
                presence="willingness=open" .../>
                <client-package>
                    <dc-cnt-metadata content-price="0.02" content-id="0xB12F" parental-rating="G" delivery-priority="1 – Low"
                    content-updated ="2009-10-1T10:30:00Z" …/>
                    <content>
                    <!– the content block is opaque for the DCD enabler and should be parsed by the application only -->
                        <x:text>"Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings during
        its key holiday quarter, topping analysts' estimates. Shares of Wal-Mart jumped nearly 3 percent" </x:text><x:image>
        "
        XhGargtWNO7PsFk90xQpqPTOrNTZ0Ux9xdvIjLpwz3"</x:image><x:ad>http://myads.com/SSXLSW&dc </x:ad>
                    </content>
                </client-package>
        </server-package>
        <server-package>
                …
        </server-package>
                …
         <channel-id>Hot News</channel-id>
          <subscription-ids>0x1765</subscription-ids>
    </ContentUpdate_overCPDE>
</dcd>
```

The XML fragment below shows an example of a well-formed ContentUpdatePush message over a DCD-2 interface (see Section 7.1.2.1). When the DCD Client receives this message, it extracts the content metadata associated with the content package and uses this metadata to process the content envelope. The DCD Client provides some of the content metadata parameters to the DCD Enabled Client Application (see Section 8.3.2).

```
<dcd>
    <ContentUpdatePush_overDCD2>
        <session-id>123abc</session-id>
        <message-id>17A78B02</message-id>
        <channel-id>Hot News</channel-id>
        <client-package>
            <dc-cnt-metadata content-price="0.02" metadata content-id="0xB12F" replaces-content-id="0xB12F" content-encoding="GZIP"
                content-updated ="2009-10-1T10:30:00Z" …/>
            <content>
                <!– the content block is opaque for the DCD enabler and should be parsed by the application only -->
                    <x:text>"Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings during
its key holiday quarter, topping analysts' estimates. Shares of Wal-Mart jumped nearly 3  percent" </x:text><x:image>
"
XhGargtWNO7PsFk90xQpqPTOrNTZ0Ux9xdvIjLpwz3"</x:image><x:ad>http://myads.com/SSXLSW&dc </x:ad>
            </content>
        </client-package>
        <client-package>
        …
        </client-package>
    …
</ContentUpdatePush_overDCD2>
</dcd>
```

The XML fragment below shows an example of a well-formed Content message over a DCD-CADE interface (see Section 7.3.2.2). When the DCD Enabled Client Application receives this message, it uses the content metadata included in the *deca-content-metadata* parameter to process the content payload embedded in the *dcd-content* element.

```
<dcd>
    <Content>
        <deca-content-metadata content-id="0xB12F" replaces-content-id="0xB12F" content-updated ="2009-10-10T10:30:00Z" …/>
        <dcd-content>
            <x:text>"Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings during  its key holiday
            quarter, topping analysts' estimates. Shares of Wal-Mart jumped nearly 3 percent"</x:text>
            <x:image>"
```

```
                    WpjQu5yqmCYsapyuvUUlvONmOZtfzgFzfagowXhGargtWNO7PsFk90xQpqPTOrNTZ0Ux9xdvIjLpwz3" </x:image>
                <x:ad>http://myads.com/SSXLSW&dc</x:ad>
        </dcd-content>
    </Content>
</dcd>
```

C.2 atom+dcd-xml Packaging Schema

The XML fragment below shows an example of a well-formed ContentUpdate message over a DCD-CPDE interface (see Section 7.2.2.3) with ATOM content packaged according to atom+dcd-xml schema. When the DCD Server receives this message, it extracts server-related content metadata associated with the content package from the *feed* element and uses this metadata to process the embedded ATOM feed. The DCD Sever may add or update DCD Client related content metadata parameters, if necessary.

```
<feed xmlns=http://www.w3.org/2005/Atom xmlns:dcd="http://www.openmobilealliance.com/oma-dcd/1.0>
        <dcd:message-type> ContentUpdate_overCPDE</dcd:message-type>
        <dcd:channel-id>Hot News</dcd:channel-id>
        <dcd:subscription-id>0x1765</dcd:subscription-id>
        <dcd:ds-cnt-metadata content-block-id="0xA017" content-id="0xB12F" replaces-content-id="0xB12F"
        deliver-at="2009-10-10T18:30:02Z" deliver-when-roaming=true
        network-preferences="WiMAX;UMTS;LTE" deliver-per-presence="willingness=open" .../>
            <dcd:dc-cnt-metadata content-price="0.02" content-id="0xB12F" replaces-content-id="0xB12F"
            content-updated ="2009-10-10T10:30:00Z" …/>
        <!– the content below is opaque for the DCD enabler and should be parsed by the application only -->
        <title>Top business stories</title>
        <link href="http://www.abc.com/bus/rss/">
        <updated>2009-01-08T11:59:55Z</updated>
        <id>urn:uuid:1724c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <author>
            <name>James Willis</name>
            <email>jwillis@abc.com</email>
        </author>
        <entry>
            <title>Wal-Mart reports jump in profit</title>
            <summary>Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings
                during its key holiday quarter, topping analysts' estimates.Shares of Wal-Mart jumped nearly 3 percent
```

```
                    </summary>
                    <id>urn:uuid:61a73c80-d399-11d9-b93C-0003939e0af6</id>
                    <updated>2007-02-21T11:59:55Z</updated>
                    <link>http://abc.com/click/here.pl?80da344efa6a</link>
                </entry>
                <entry>
                    …
                </entry>
        </feed>
        <feed>
            …
        </feed>
        …
```

The XML fragment below shows an example of a well-formed ContentUpdatePush message over the DCD-2 interface (see Section 7.1.2.1) with ATOM content packaged according to atom+dcd-xml schema. When the DCD Client receives this message, it extracts content metadata associated with the content package from the *feed* element and uses this metadata to process the embedded ATOM feed. The DCD Client provides some of the content metadata parameters to the DCD Enabled Client Application (see Section 8.3.2).

```
<feed xmlns=http://www.w3.org/2005/Atom xmlns:dcd="http://www.openmobilealliance.com/oma-dcd/1.0>
        <dcd:message-type> ContentUpdatePush_overDCD2</dcd:message-type>
        <dcd:session-id>123abc</dcd:session-id>
        <dcd: message-id>17A78B02</dc: message-id>
        <dcd:channel-id>Hot News</dcd:channel-id>
        <dcd:dc-cnt-metadata content-price="0.02" content-id="0xB12F" replaces-content-id="0xB12F"
        content-encoding="GZIP" content-updated ="2009-10-10T10:30:00Z" …/>
    <!– the content below is opaque for the DCD enabler and should be parsed by the application only -->
        <title>Top business stories</title>
        <link href="http://www.abc.com/bus/rss/">
        <updated>2009-01-08T11:59:55Z</updated>
        <id>urn:uuid:1724c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <author>
            <name>James Willis</name>
            <email>jwillis@abc.com</email>
        </author>
        <entry>
```

```
            <title>Wal-Mart reports jump in profit</title>
            <summary>Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings
        during its key holiday quarter, topping analysts' estimates.Shares of Wal-Mart jumped nearly 3 percent
            </summary>
            <id>urn:uuid:61a73c80-d399-11d9-b93C-0003939e0af6</id>
            <updated>2007-02-21T11:59:55Z</updated>
            <link>http://abc.com/click/here.pl?80da344efa6a</link>
        </entry>
        <entry>
          …
        </entry>
    </feed>
    <feed>
      …
    </feed>
      …
```

The XML fragment below shows an example of a well-formed Content message over DCD-CADE interface (see Section 7.3.2.2) with ATOM feed as a content payload. When the DCD Enabled Client Application receives this message, it uses the content metadata included in *deca-content-metadata* parameter to process the ATOM feed embedded in the *dcd-content* element.

```
    <feed xmlns=http://www.w3.org/2005/Atom xmlns:dcd="http://www.openmobilealliance.com/oma-dcd/1.0>
        <dcd:message-type> Content</dcd:message-type>
        <dcd:channel-id>Hot News</dcd:channel-id>
        <dcd:deca-content-metadata content-id="0xB12F" replaces-content-id="0xB12F"
                                   content-updated ="2009-1-1T10:30:00Z" …/>
        <title>Top business stories</title>
        <link href="http://www.abc.com/bus/rss/">
        <updated>2009-01-08T11:59:55Z</updated>
        <id>urn:uuid:1724c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <author>
            <name>James Willis</name>
            <email>jwillis@abc.com</email>
        </author>
        <entry>
            <title>Wal-Mart reports jump in profit</title>
```

```
            <summary>Wal-Mart Stores, the world's largest retailer, on Tuesday reported a jump in earnings
                during its key holiday quarter, topping analysts' estimates.Shares of Wal-Mart jumped nearly 3 percent
            </summary>
            <id>urn:uuid:61a73c80-d399-11d9-b93C-0003939e0af6</id>
            <updated>2007-02-21T11:59:55Z</updated>
            <link>http://abc.com/click/here.pl?80da344efa6a</link>
        </entry>
        <entry>
        …
        </entry>
</feed>
<feed>
…
</feed>
…
```

Appendix D. UAProf Schema for DCD Characteristics

This section is normative.

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:prf="http://www.openmobilealliance.org/tech/profiles/ccppschema-dcd-v1_0.rdf">
<rdf:Description rdf:ID="Component">
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
<rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdfschema# Resource" />
<rdfs:label>Component</rdfs:label>
<rdfs:comment>A Component within the CC/PP Schema is a class of related properties that describe the
capabilities and preferences information.</rdfs:comment>
</rdf:Description>
<!--  ************************************************************   -->
<!--  ***** Properties shared among the components*****   -->
<rdf:Description rdf:ID="component">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdfs:label>component</rdfs:label>
<rdfs:comment>The component attribute links the various components to the root
node(profile).</rdfs:comment>
</rdf:Description>
<!--  ************************************************************   -->
<!--  ***** Component Definitions *****   -->
<rdf:Description rdf:ID="DcdCharacteristics">
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
<rdfs:subClassOf rdf:resource="#Component" />
<rdfs:label>Component: DcdCharacteristics</rdfs:label>
<rdfs:comment>The DcdCharacteristics component contains properties of the device's Dynamic Content
Delivery capabilities, such as bearers and protocols supported, authentication methods,
etc.</rdfs:comment>
</rdf:Description>
<!--  ************************************************************   -->
<!--  ***** Component: DcdCharacteristics *****   -->
<!--  ************************************************************   -->
<!--  ***** Attributes for component: DcdCharacteristics *****   -->
<rdf:Description rdf:ID="DcdClientName">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: Name of the DCD Client, in vendor / client / version format. Type: Literal
Resolution: Locked Examples: "SPARQY/DC-Leet/1.0"</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:ID="DcdVersion">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: DCD versions supported by the DCD Client, conveyed as
majorVersionNumber.minorVersionNumber. Type: Literal bag Resolution: Locked Examples: "1.0",
"1.1"</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:ID="DcdSupportedBearers">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: Transport bearers for DCD that are supported by the device. Type: Literal bag
Resolution: Locked Examples: "UMTS", "WiMAX", "LTE", "802.11", "BCAST", "CBS"</rdfs:comment>
</rdf:Description>
```

```
<rdf:Description rdf:ID="DcdSupportedProtocols">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: Transport protocols for DCD that are supported by the device. Type: Literal
bag Resolution: Locked Examples: "HTTP", "HTTPS"</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:ID="DcdSupportedMethods">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: Content delivery methods supported by the DCD Client. Type: Literal bag
Resolution: Locked Examples: "Pull", "Push"</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:ID="DcdAuthMethods">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: DCD session authentication methods supported by the device. Type: Literal bag
Resolution: Locked Examples: "digest-user", "digest-gba", "x509"</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:ID=" DcdAcceptSchema ">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />
<rdfs:domain rdf:resource="#DcdCharacteristics" />
<rdfs:comment>Description: DCD packaging schema supported by the device. Type: Literal bag Resolution:
Locked Examples: "dcd-xml", "atom+dcd-xml", "rss+dcd-xml"</rdfs:comment>
</rdf:Description>
</rdf:RDF>
```