



Advancing storage &  
information technology

# Cloud Data Management Interface Extension: OVF

## Version 1.0e

"Publication of this Working Draft for review and comment has been approved by the Cloud Storage Technical Working Group. This draft represents a "best effort" attempt by the Cloud Storage Technical Working Group to reach preliminary consensus, and it may be updated, replaced, or made obsolete at any time. This document should not be used as reference material or cited as other than a 'work in progress.' Suggestion for revision should be directed to <http://snia.org/feedback>."

## *Working Draft*

## Revision History

Date	Version	By	Comments
12/14/2011	1.0a	Mark Carlson, Oracle	Document created
1/16/2012	1.0b	Marie McMinn	Updates to include standard SNIA front matter and technical edit
1/20/2012	1.0c	Mark Carlson, Oracle	Split OVF and CIMI extensions
1/26/2012	1.0d	Mark Carlson, Oracle	Changed CDMI attributes to CDMI fields per face to face discussion
1/26/2012	1.0e	Marie McMinn	Includes minor edits.

The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

- Any text, diagram, chart, table, or definition reproduced shall be reproduced in its entirety with no alteration, and,
- Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced shall acknowledge the SNIA copyright on that material, and shall credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, sell any excerpt or this entire document, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by e-mailing [tcmd@snia.org](mailto:tcmd@snia.org). Please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

Copyright © 2012 Storage Networking Industry Association.

## OVF CDMI Extension

The DMTF OVF Standard (DSP 243) is used to move virtual appliances between clouds. This extension to CDMI supports both of these standards such that CDMI can be used with these standards to manage data and storage for IaaS clouds.

### Modifications to the current CDMI spec:

The changes are as follows:

- Support for de-serializing an OVF package into CDMI containers and objects such that they can be referenced by CIMI machine images and volume images

#### 1) Insert into Clause 3 - "Terms":

##### 3.x

##### Open Virtualization Format

##### OVF

a DMTF standard for virtual appliance portability

##### 3.x

##### OVF package

an OVF descriptor file accompanied by zero or more files

##### 3.x

##### OVF descriptor

an OVF file with XML metadata describing a virtual appliance and its requirements on the environment

#### 2) Add a table entry to the end of Table 102 in Clause 12.1.1 as follows:

Capability Name	Type	Definition
cdmi_ovf_support	JSON String	If present and "true", this capability indicates that the cloud storage system supports OVF serialization and deserialization per 15.4 OVF Support.

#### 3) Add a table entry to the end of Table 106 in Clause 12.1.5 as follows:

Capability Name	Type	Definition
cdmi_ovf_support	JSON String	If present and "true", this capability indicates that the container supports OVF serialization and deserialization per 15.4 OVF Support.

#### 4) Insert new Clause after 15.3 Importing Serialized Data:

### 15.4 OVF Support

The DMTF Open Virtualization Format (OVF) standard allows a virtual appliance to be serialized from one platform and deserialized onto another platform. CDMI supports OVF packages that are bundled into a TAR file as an optional serialization format from that defined above. An OVF package can be created as a CDMI object by using the TAR file as the value of the object. The

object's mimetype field shall be set to "application/ovf" to indicate that the object contains an OVF package.

To expand an OVF package from the CDMI object, the object is used to create a new container, setting the deserialize field to the URI of the OVF package CDMI object.

As a result, a new container will be created with the following children:

- a CDMI object with the value set to the OVF descriptor contents;
- a CDMI object with the value set to the OVF manifest file contents;
- a CDMI object with the value set to the certificate file contents;
- CDMI objects for any other files in the OVF package that are not virtual disk contents; and
- a CDMI container for each disk in the disk section of the OVF descriptor metadata.

The CDMI containers that are created in bullet five are meant to be exported to virtual machines with a CIMI or OCCI exported protocol. These containers shall have the contents for access by these protocols initialized from the files specified by the `ovf:fileRef` attribute in the corresponding disk section of the OVF descriptor (see 9.1 of the OVF standard). If `ovf:fileRef` is omitted, the container has no initial contents as seen by the exported protocol.

The container name for each of these containers shall be set to the `ovf:diskId` attribute in the corresponding disk section of the OVF descriptor.

The container's `cdmi_assignedsize` metadata value shall be set to the value of the `ovf:capacity` attribute in the corresponding disk section of the OVF descriptor.

As described above, to serialize an OVF package from a container back into a TAR file version, 1) create a CDMI object, 2) set the `serialize` field to the URI of the container, and 3) set the `mimetype` field to "application/ovf". The OVF serialization process uses the OVF package manifest file to understand how to serialize the package and to understand which objects and containers to serialize as per the OVF standard. The resultant object will be a TAR file version of the OVF package.