

# OpenCable Application Platform Specifications OCAP Extensions

## OCAP Home Networking Extension

### OC-SP-OCAP-HNEXT-I05-100603

**ISSUED**

#### **Notice**

This OpenCable specification is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein.

© Copyright 2005-2010 Cable Television Laboratories, Inc.  
All rights reserved.

## Document Status Sheet

<b>Document Control Number:</b>	OC-SP-OCAP-HNEXT-I05-100603			
<b>Document Title:</b>	OCAP Home Networking Extension			
<b>Revision History:</b>	I01 – Released 5/19/05			
	I02 – Released 12/20/07			
	I03 – Released 4/18/08			
	I04 – Released 12/17/09			
	I05 – Released 6/3/10			
<b>Date:</b>	June 3, 2010			
<b>Status:</b>	<del>Work in Progress</del>	<del>Draft</del>	<b>Issued</b>	<del>Closed</del>
<b>Distribution Restrictions:</b>	<del>Author Only</del>	<del>CL/Member</del>	<del>CL/Member/Vendor</del>	<b>Public</b>

### Key to Document Status Codes

- Work in Progress** An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
- Draft** A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
- Issued** A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
- Closed** A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

### Trademarks

CableLabs<sup>®</sup>, DOCSIS<sup>®</sup>, EuroDOCSIS<sup>™</sup>, eDOCSIS<sup>™</sup>, M-CMTS<sup>™</sup>, PacketCable<sup>™</sup>, EuroPacketCable<sup>™</sup>, PCMM<sup>™</sup>, CableHome<sup>®</sup>, CableOffice<sup>™</sup>, OpenCable<sup>™</sup>, OCAP<sup>™</sup>, CableCARD<sup>™</sup>, M-Card<sup>™</sup>, DCAS<sup>™</sup>, tru2way<sup>™</sup>, and CablePC<sup>™</sup> are trademarks of Cable Television Laboratories, Inc.

# Contents

<b>1</b>	<b>SCOPE</b>	<b>1</b>
1.1	OCAP HOME NETWORKING EXTENSION PURPOSE	1
1.2	OCAP HOME NETWORKING EXTENSION REQUIREMENTS	1
1.3	OCAP HOME NETWORKING ENVIRONMENT	1
1.4	OCAP HOME NETWORKING APPLICATION AREAS (INFORMATIVE)	2
1.4.1	<i>Connected Device and Content Discovery</i>	2
1.4.2	<i>Presenting Content</i>	2
1.4.3	<i>Remote Recording</i>	3
1.4.4	<i>Network Recording Requests</i>	3
<b>2</b>	<b>REFERENCES</b>	<b>4</b>
2.1	NORMATIVE REFERENCES	4
2.2	INFORMATIVE REFERENCES	4
2.3	REFERENCE ACQUISITION	4
<b>3</b>	<b>GLOSSARY</b>	<b>5</b>
<b>4</b>	<b>ACRONYMS</b>	<b>7</b>
<b>5</b>	<b>CONVENTIONS</b>	<b>8</b>
5.1	SPECIFICATION LANGUAGE	8
5.2	ORGANIZATION	8
<b>6</b>	<b>OCAP HOME NETWORKING</b>	<b>9</b>
6.1	INTRODUCTION	9
6.2	DEVICE DISCOVERY	9
6.2.1	<i>Device Discovery Process</i>	10
6.3	CONTENT DISCOVERY AND LISTING	10
6.3.1	<i>Content Discovery and Listing Process</i>	10
6.3.2	<i>Content Searching</i>	10
6.3.3	<i>Content Metadata Caching</i>	10
6.4	RECORDING OPERATIONS	11
6.4.1	<i>Remote Recording</i>	11
6.4.2	<i>Network Recording Request Management</i>	11
6.5	PERMISSIONS	11
6.5.1	<i>Unsigned Applications</i>	11
6.5.2	<i>Signed Applications</i>	12
6.5.3	<i>Monitor Applications</i>	12
6.6	API SUPPORT PROPERTY	12
6.7	OCAP HOME NETWORKING API	12
6.7.1	<i>Additions to OCAP Packages</i>	12
6.7.2	<i>OCAP Home Networking Packages</i>	12
<b>7</b>	<b>SECURITY</b>	<b>14</b>
7.1	HOME NETWORK PERMISSION	14
7.2	MONITOR APPLICATION PERMISSION	14
<b>8</b>	<b>REGISTRY OF CONSTANTS</b>	<b>15</b>
<b>ANNEX A</b>	<b>HOME NETWORKING API</b>	<b>23</b>

PACKAGE ORG.OCAP.HN ..... 23

ORG.OCAP.HN CLASS CONTENTSERVEREVENT ..... 24

ORG.OCAP.HN INTERFACE CONTENTSERVERLISTENER ..... 27

ORG.OCAP.HN INTERFACE CONTENTSERVERNETMODULE ..... 28

ORG.OCAP.HN INTERFACE DEVICE ..... 32

ORG.OCAP.HN CLASS DEVICEEVENT ..... 43

ORG.OCAP.HN INTERFACE DEVICEEVENTLISTENER ..... 46

ORG.OCAP.HN CLASS HOMENETPERMISSION ..... 47

ORG.OCAP.HN CLASS NETACTIONEVENT ..... 49

ORG.OCAP.HN INTERFACE NETACTIONHANDLER ..... 52

ORG.OCAP.HN INTERFACE NETACTIONREQUEST ..... 53

ORG.OCAP.HN INTERFACE NETLIST ..... 55

ORG.OCAP.HN CLASS NETMANAGER ..... 57

ORG.OCAP.HN INTERFACE NETMODULE ..... 61

ORG.OCAP.HN CLASS NETMODULEEVENT ..... 65

ORG.OCAP.HN INTERFACE NETMODULEEVENTLISTENER ..... 68

ORG.OCAP.HN CLASS NETWORKINTERFACE ..... 69

ORG.OCAP.HN CLASS NOTAUTHORIZEDEXCEPTION ..... 72

ORG.OCAP.HN CLASS PROPERTYFILTER ..... 74

**ANNEX B CONTENT API..... 76**

PACKAGE ORG.OCAP.HN.CONTENT ..... 76

ORG.OCAP.HN.CONTENT INTERFACE AUDIORESOURCE ..... 77

ORG.OCAP.HN.CONTENT INTERFACE CONTENTCONTAINER ..... 79

ORG.OCAP.HN.CONTENT INTERFACE CONTENTENTRY ..... 89

ORG.OCAP.HN.CONTENT INTERFACE CONTENTITEM ..... 92

ORG.OCAP.HN.CONTENT INTERFACE CONTENTRESOURCE ..... 98

ORG.OCAP.HN.CONTENT CLASS DATABASEEXCEPTION ..... 102

ORG.OCAP.HN.CONTENT INTERFACE IOSTATUS ..... 105

ORG.OCAP.HN.CONTENT CLASS METADATAIDENTIFIERS ..... 106

ORG.OCAP.HN.CONTENT CLASS METADATANODE ..... 108

ORG.OCAP.HN.CONTENT INTERFACE STREAMABLECONTENTRESOURCE ..... 113

ORG.OCAP.HN.CONTENT INTERFACE VIDEORESOURCE ..... 114

**ANNEX C CONTENT NAVIGATION API ..... 115**

PACKAGE ORG.OCAP.HN.CONTENT.NAVIGATION ..... 115

ORG.OCAP.HN.CONTENT.NAVIGATION CLASS CONTENTDATABASEFILTER ..... 116

ORG.OCAP.HN.CONTENT.NAVIGATION INTERFACE CONTENTLIST ..... 117

ORG.OCAP.HN.CONTENT.NAVIGATION CLASS DATABASEQUERY ..... 120

ORG.OCAP.HN.CONTENT.NAVIGATION CLASS DEVICEFILTER ..... 125

**ANNEX D UPNP PROFILES API..... 126**

PACKAGE ORG.OCAP.HN.PROFILES.UPNP ..... 126

ORG.OCAP.HN.PROFILES.UPNP INTERFACE UPNPCONSTANTS ..... 127

**ANNEX E SERVICE API..... 139**

PACKAGE ORG.OCAP.HN.SERVICE ..... 139

ORG.OCAP.HN.SERVICE INTERFACE REMOTESERVICE ..... 140

**ANNEX F RECORDING API ..... 141**

PACKAGE ORG.OCAP.HN.RECORDING ..... 141

ORG.OCAP.HN.RECORDING INTERFACE NETRECORDINGENTRY ..... 142

ORG.OCAP.HN.RECORDING INTERFACE NETRECORDINGREQUESTHANDLER ..... 144

ORG.OCAP.HN.RECORDING INTERFACE NETRECORDINGREQUESTMANAGER ..... 147

ORG.OCAP.HN.RECORDING CLASS NETRECORDINGSPEC ..... 149

ORG.OCAP.HN.RECORDING INTERFACE RECORDINGCONTENTITEM .....151  
ORG.OCAP.HN.RECORDING INTERFACE RECORDINGNETMODULE.....157  
**ANNEX G SECURITY API.....161**  
PACKAGE ORG.OCAP.HN.SECURITY.....161  
ORG.OCAP.HN.SECURITY INTERFACE NETAUTHORIZATIONHANDLER.....162  
ORG.OCAP.HN.SECURITY CLASS NETSECURITYMANAGER.....164  
**APPENDIX I REVISION HISTORY .....168**

## Figures

FIGURE 1-1 - DIGITAL HOME ENTERTAINMENT SYSTEM .....2

## Tables

TABLE 6-1 - API SUPPORT PROPERTY .....12

This page intentionally left blank.

# 1 SCOPE

This document defines a minimal profile specification for a Home Networking software environment for digital cable receivers, with or without local storage. This platform is a modular extension to the OpenCable Application Platform 1.1 [OCAP]. The OCAP Specification and the OCAP Home Networking Extension Specification were developed by Cable Television Laboratories, Inc. (CableLabs ) in conjunction with representatives from a number of its member cable operating companies, as well as leading software and hardware firms.

The OCAP Home Networking Extensions Specification is based on the OCAP 1.1 Profile and includes that Specification in its entirety.

## 1.1 OCAP Home Networking Extension Purpose

OCAP Home Networking Extension specification describes an application interface that includes all required Application Program Interfaces (APIs), content and data formats, and protocols up to the application level. Applications developed to the OCAP Home Networking Extension (HN) Specification will be executed on OpenCable-compliant Host devices. The OCAP HN Specification allows cable operators to deploy their applications and services on all OpenCable-compliant host devices connected to their networks.

The OCAP Home Networking platform SHALL be applicable to a wide variety of hardware and operating systems to allow Consumer Electronics (CE) manufacturers flexibility in implementation. A primary objective in defining OCAP Home Networking is to enable competing implementations of the OCAP Home Networking platform by CE manufacturers.

## 1.2 OCAP Home Networking Extension Requirements

The OCAP Home Networking platform has been designed to meet specific requirements that are not commonly applied to other Home Networking environments. Some of these requirements are related to content protection obligations that cable operators face, to the fact that broadcast event descriptions might be maintained by applications, and to the fact that the platform supports DVR applications deployed by different service providers that have no *a priori* knowledge of each other and, therefore, may compete for resources.

## 1.3 OCAP Home Networking Environment

The OCAP Home Networking extensions have been designed to provide OCAP home networking applications access to digital entertainment content available on devices connected to a cable service subscriber's home network. This includes content stored on the home networking-capable OpenCable Host device hosting the applications, as well as on other home networking-capable OpenCable Hosts and non-OpenCable Host devices connected to the home network. The OCAP Home Networking APIs assume that the Host and at least some of the other devices connected to the home network implement a set of protocols supporting the following functions:

- Advertisement and discovery of connected devices
- Advertisement and discovery of digital entertainment content stored on devices
- Communication of information about a device's capability to serve and/or receive digital entertainment content
- Communication of information necessary to establish the transfer of digital entertainment content across the home network

- Communication of playback control, file operations, and other commands needed to support the user's enjoyment of digital entertainment content

If these protocols are not implemented by two or more devices connected to the home network, features supported by the OCAP Home Networking extensions are not useable. Specification of the protocols supporting the functions described above is out of scope for the OCAP Home Networking Extension specification.

Figure 1–1 illustrates a possible home network environment in which the OpenCable Host, implementing OCAP Home Network Extensions, can share digital entertainment content with other devices connected to the home network.

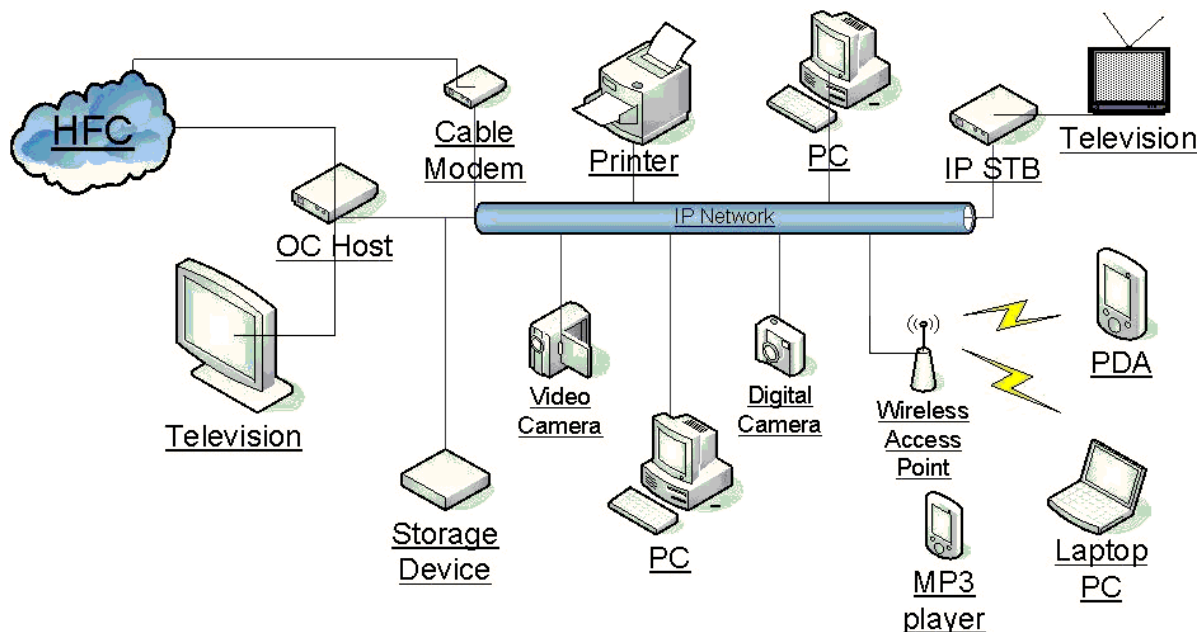


Figure 1–1 - Digital Home Entertainment System

## 1.4 OCAP Home Networking Application Areas (informative)

The information in this section is informative to the OCAP Home Networking Extensions Specification.

This section identifies the applications and services that could be made available to the viewer when using an OCAP Home Networking-compliant terminal. The descriptions of the applications are intended to demonstrate the scope of services required from OCAP.

### 1.4.1 Connected Device and Content Discovery

A critical application enabled by this platform is the discovery of connected devices and/or their audio or video content. Discovery of content available on the home network and its presentation to the user is the first step in setting up a transfer of content from a source (server) device to a rendering device, over a home network.

### 1.4.2 Presenting Content

Presentation of discovered multimedia content to the user offers ample opportunity for application innovation. How the content listing is organized, how it is displayed, how an item is selected, options offered for filtering and

managing it, and presentation of metadata are some of the ways an application can differentiate itself and provide value for the end user.

### **1.4.3 Remote Recording**

On set-top boxes that implement both home networking and DVR functionality, requests for scheduling recordings may originate from devices on the home network. Applications may receive, prioritize, and ultimately resolve these network requests to recordings on the local device.

### **1.4.4 Network Recording Requests**

On a user's home network, there may be multiple devices that support DVR functionality. Applications may allow users to schedule DVR recordings on these devices from a single set-top box. Applications may also allow users to browse the existing schedule of recordings across all devices on the home network.

## 2 REFERENCES

This section provides the normative and informative references used to create this specification.

### 2.1 Normative References

**Note:** Information contained in these normative references is required for all implementations. Notwithstanding, intellectual property rights may be required to use or implement these normative references.

The following documents contain provisions which, through reference in this text, constitute provisions of the present document. References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific:

- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

References	Edition	Description
[HOST]	v2.1	OpenCable Host Device 2.1 Core Functional Requirements, OC-SP-HOST2.1-CFR- I11-100507, May 7, 2010, Cable Television Laboratories, Inc.
[OCAP]	OCAP 1.1	OpenCable Application Platform Specification, OCAP 1.1 Profile, OC-SP-OCAP1.1.3-100603, June 3, 2010, Cable Television Laboratories, Inc.
[UPnP CDS]	June 25, 2002	UPnP ContentDirectory v3, ContentDirectory:3 Service Template Version 1.01, UPnP Forum, September 30, 2008.

### 2.2 Informative References

None.

### 2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; <http://www.cablelabs.com>
- UPnP Forum, [www.upnp.org](http://www.upnp.org)

### 3 GLOSSARY

The following definitions are used in this specification:

<b>Connected Device</b>	A physical device that is currently connected to a communication network and is compliant with protocols specified or referred to in this specification, enabling it to be discoverable by OCAP Home Networking applications through OCAP Home Networking APIs.
<b>Content</b>	Streaming media, file-based media, applications, and data, including metadata, that are discoverable by OCAP applications through OCAP Home Networking APIs.
<b>Content Resource Properties</b>	Information about content such as file name, file size, date created, and metadata.
<b>Content Usage Rules</b>	As defined in CHILA and DFAST
<b>Device Capability</b>	The ability of a given device to perform a specific function, such as server, data store, or renderer.
<b>Device Property</b>	A specific attribute of a given device.
<b>Device Type</b>	As defined in the XML Device Schema acquired via the discovery process or as defined in the CableLabs OpenCable Host specification.
<b>Discovery</b>	The process by which OCAP HN applications find, directly or by proxy, devices and content that are exposed to OCAP applications.
<b>Home Network</b>	A communication system over which Connected Devices may be discovered and content shared.
<b>Legacy Device</b>	A set-top box with embedded security.
<b>OCAP Home Network</b>	<p>An Internet Protocol (IP)-based communication network in a cable television service subscriber's home that allows the discovery and sharing of content among devices capable of storing, receiving or transferring data within the confines of the subscriber's local area network (LAN) itself. The OCAP Home Network does not support or allow the transfer of copy-protected content beyond the physical boundaries of the subscriber's home LAN.</p> <p>The OCAP Home Network extensions build upon and references underlying protocols and other capabilities defined (or to be defined) by the OpenCable specifications and/or other CableLabs specifications.</p> <p>The physical network media are irrelevant to specification of the OCAP Home Networking extensions. Although this specification is largely independent of the underlying network protocols, references to those protocols may appear in portions of this specification.</p>
<b>OCAP Home Network Application</b>	An OCAP application, as defined in [OCAP], designed to use OCAP Home Networking APIs to provide the cable subscriber value by enabling the sharing of digital entertainment content stored on Connected Devices, including but not limited to providing the following services: discovering Connected Devices, discovering content stored on Connected Devices, organizing and presenting to a user information about content stored on Connected Devices, and directing content on one Connected Device to be transferred to and rendered on another Connected Device.
<b>OCAP Home Network Implementation</b>	A hardware and software environment that conforms to the OCAP Home Networking Extensions specification.

**OCAP Home Network  
Platform**

The hardware, software, policies, and requirements definitions embodied in the OCAP HN and related specifications.

**Rendering**

Display of coded content (audio, video, or still image).

## 4 ACRONYMS

The following acronyms are used in this specification:

<b>API</b>	Application Programming Interface
<b>CCI</b>	Copy Control Information
<b>CHILA</b>	CableLabs CableCARD-Host Interface License Agreement
<b>DFAST</b>	Dynamic Feedback Arrangement Scrambling Technique
<b>DLNA</b>	Digital Living Network Alliance
<b>DVR</b>	Digital Video Recorder
<b>HN</b>	Home Network or Home Networking
<b>ISO</b>	International Organization for Standardization
<b>IETF</b>	Internet Engineering Task Force
<b>LAN</b>	Local Area Network
<b>OCAP</b>	OpenCable Application Platform
<b>PVR</b>	Personal Video Recorder
<b>RFC</b>	Request for Comments
<b>UPnP</b>	Universal Plug and Play
<b>URI</b>	Uniform Resource Identifier

## 5 CONVENTIONS

The following conventions are used in this manual:

- The following font type is used to indicate code examples, names of properties, and other information that **MUST** be entered exactly as-is: `code example font`
- **Boldfaced** text is used as emphasis.

### 5.1 Specification Language

The following words are used throughout this document to define the significance of particular requirements:

SHALL	This word means that the item is an absolute requirement of this specification.
SHALL NOT	This phrase means that the item is an absolute prohibition of this specification.
SHOULD	This word means that valid reasons may exist in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
SHOULD NOT	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
MAY	this word, or the adjective <b>OPTIONAL</b> , means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

Other text is descriptive or explanatory.

### 5.2 Organization

This document uses [OCAP] as its base. Where applicable, OCAP Home Networking sections reference the corresponding section within [OCAP].

The `org.ocap.hn` API package is defined in Annex A.

The `org.ocap.hn.content` API package is defined in Annex B.

The `org.ocap.hn.content.navigation` API package is defined in Annex C.

The `org.ocap.hn.profiles.upnp` API package is defined in Annex D.

The `org.ocap.hn.service` API package is defined in Annex E.

The `org.ocap.hn.recording` API package is defined in Annex F.

The `org.ocap.hn.security` API package is defined in Annex G.

## 6 OCAP HOME NETWORKING

This chapter describes the OCAP Home Networking (HN) platform. The OCAP HN APIs are defined and described in more detail in Annex A, Annex B, Annex C, Annex D, Annex E, Annex F, and Annex G.

### 6.1 Introduction

This specification fully defines a Home Networking extension to [OCAP]. Implementers of this specification SHALL also fully implement [OCAP]. [OCAP] and this Home Networking extension platform are related in such a way that [OCAP] implementations have no build-time or run-time dependencies on the Home Networking extensions, while OCAP Home Network Implementations depend on the full implementation of [OCAP].

An OpenCable Host that implements the OCAP Home Networking extension MAY implement the OCAP DVR extension as well.

OCAP Home Networking extensions rely upon support in the OCAP Home Network Implementation and in Connected Devices of a set of compatible, IP-based device and Content Discovery LAN communication protocols to expose information about the Connected Devices, their capabilities, and the digital entertainment Content they store, if any. If the hardware and software platform that implements the OCAP Home Networking APIs and Connected Devices do not implement these protocols, then information about the devices, their capabilities, and stored Content is not conveyed to the OCAP Home Network Application through the implementation and the OCAP Home Networking APIs. Connected Devices that implement device and Content Discovery LAN communication protocols compliant with protocols supported by the OCAP Home Network implementation, are referred to as protocol-compliant devices. Definition of the IP-based device and Content Discovery LAN communication protocols is outside the scope of this OCAP Home Networking Extension specification.

IP-based device and Content Discovery LAN communication protocols are specified in the OpenCable Host 2.0 Home Networking Extension specification.

It is assumed that Content stored on the OpenCable Host, using OCAP DVR functionality, could be exposed to protocol compliant devices sharing the IP-based communication network, and Content exposed by protocol compliant devices sharing the IP-based communication network, could be moved or copied to DVR Content storage resources on the OpenCable Host device.

This specification defines a platform to enable applications to discover devices on a Home Network, gain listings of Content resources on Connected Devices, and perform certain operations on Content resources, such as Rendering, copying, and deleting. These functions are considered basic capabilities of a home networking platform. Advanced functions, such as device-specific operations, are not supported by this platform. This chapter describes the platform in a textual format, and places normative requirements on implementations. Annexes to this specification define the platform APIs and contain detailed API descriptions and normative statements.

The OCAP Home Networking APIs assume that the OpenCable Host platform, in which they are implemented, provides an interface to an Internet Protocol (IP)-based communication network in a cable service subscriber's premises.

### 6.2 Device Discovery

OCAP Home Network Applications are provided with the ability to detect the presence of Connected Devices on the OCAP Home Network, discover the properties and capabilities of those devices, and detect changes in the status

of those devices. Device information acquired by the OCAP Home Network Implementation is exposed to OCAP applications, in particular to OCAP Home Network Applications, through the OCAP Home Networking APIs.

### 6.2.1 Device Discovery Process

The process used by the OCAP Home Network Implementation for discovering devices connected on the Home Network includes the following activities:

- a) Maintaining a list of Connected Devices.
- b) Maintaining metadata for each Connected Device, such that its type, sub-devices, capabilities, properties, and supported functionalities may be exposed to applications.
- c) Generating application events when devices are added or removed from the list of Connected Devices.

## 6.3 Content Discovery and Listing

OCAP Home Networking APIs provide OCAP Home Network Applications with the ability to detect the presence of Content resources accessible on devices connected on the OCAP Home Network. Content information acquired by the OCAP Home Network Implementation is exposed to OCAP applications through the OCAP Home Networking APIs.

### 6.3.1 Content Discovery and Listing Process

The process for discovering Content on Connected Devices on the Home Network includes the following activities:

- a) Generating, on application request, a list of Content resources available on a Connected Device.
- b) Generating, on application request, a list of Content resources available on all Connected Devices.
- c) Generating, on application request, a list containing a subset of Content resources available on a Connected Device, where the subset is defined by application specified search criteria.
- d) Generating, on application request, metadata associated with Content resource.
- e) Generating application events when Content resources are added, removed, or modified from a list of Content resources.

### 6.3.2 Content Searching

The ability to perform detailed searches for Content resources accessible on devices connected on the OCAP Home Network is exposed to OCAP applications through the OCAP Home Networking APIs. The search criteria used by an application to perform such searches SHALL follow the format defined by [UPnP CDS] section 2.5.5.1: Search Criteria String Syntax. Usage of the search string SHALL follow the usage specified by [UPnP CDS] section 2.5.5.2: Search Criteria String Semantics and Examples.

### 6.3.3 Content Metadata Caching

Metadata associated with any content entry that has been retrieved from the home network due to application request SHALL remain cached in local memory, at a minimum, for the lifetime of the Java object representing the entry. The home networking implementation SHALL NOT implicitly update such metadata unless directed to do so by application request. Accessing and inspection of metadata associated with a content entry SHALL NOT cause network activity. Any metadata cached in local memory as the result of an application browse or search operation SHALL be available to all applications with the necessary permissions to access home networking content.

## 6.4 Recording Operations

OCAP Home Networking applications are provided with the ability to perform a number of operations effecting scheduled future recordings and recorded Content items stored on devices connected to the OCAP Home Network, including recording schedule management operations, and Rendering.

### 6.4.1 Remote Recording

Remote Recording is the scheduling and management of DVR recordings on a Connected Device. The OCAP home networking APIs allow privileged applications to request that recordings hosted on a Connected Device be scheduled, deleted, prioritized, disabled, or rescheduled. It is up to the implementation of the Connected Device to determine whether and how such requests are honored.

### 6.4.2 Network Recording Request Management

Privileged applications may register themselves to be notified of requests for recording operations from Connected Devices on the home network. The home networking implementation SHALL notify registered monitor applications of recording operation requests originating from the OCAP Home Network. Behavior of the OCAP Home Networking implementation in response to network requests for recording operations when no request handler has been registered by any monitor application is implementation-dependent.

Included in the responsibilities of a registered recording request handler are:

- a) Handling network requests to schedule recordings on the local device.
- b) Handling network requests to modify or reschedule existing recordings on the local device.
- c) Handling network requests to stop or cancel existing recordings on the local device.
- d) Handling network requests to delete content associated with existing recordings on the local device.
- e) Handling network requests to delete metadata associated with existing recordings on the local device.
- f) Handling network requests to prioritize scarce resource allocation among existing recordings on the local device.

## 6.5 Permissions

The Java security model includes the means to restrict access to an API and to applications that have been requested and granted specific permissions. The OCAP Home Network Extension defines `HomeNetPermission`, a home networking specific permission used to control access to certain home networking APIs.

In addition to the `HomeNetPermission` class, access to content on the home network is restricted through file access permissions as defined by `org.ocap.storage.ExtendedFileAccessPermission`. Access permissions can be retrieved using the `ContentEntry.getExtendedFileAccessPermissions()` method. Access permissions MAY be signaled through the home networking content discovery protocol. In the event that metadata associated with a `ContentEntry` does not include access permissions, the OCAP Home Network Implementation SHALL associate an `ExtendedFileAccessPermission` granting full access with the `ContentEntry`.

### 6.5.1 Unsigned Applications

Unsigned applications SHALL NOT be granted access to a Home Network via the HN API in the `org.ocap.hn` name space.

## 6.5.2 Signed Applications

Signed applications MAY perform home networking actions, via the OCAP Home Networking API, in the org.ocap.hn name space. Some of the method calls in the OCAP Home Networking API require org.ocap.hn.HomeNetPermission, which MUST be provided by the applications permission request file. See Section 7.1.

## 6.5.3 Monitor Applications

Monitor Application permissions are detailed in Section 7.2.

## 6.6 API Support Property

OCAP Hosts that support the OCAP HN extension SHALL indicate this support with the system properties as defined in section 13.3.12.2 of [OCAP] per Table 6–1.

*Table 6–1 - API Support Property*

Property	Description	Value	Application Access
ocap.api.option.hn	System property indicating that OCAP HN extension is supported by the Host device	“3.0”	signed and unsigned

## 6.7 OCAP Home Networking API

The OCAP Home Network Platform extends the OCAP-J API defined in [OCAP]. The additional packages, classes, and interfaces are listed here. For a complete definition and description of the APIs, see Annex A through Annex E. An implementation of the OCAP Home Network Platform SHALL include all of the packages, classes, and interfaces defined in [OCAP], as well as the following packages, classes, and interfaces.

### 6.7.1 Additions to OCAP Packages

The OCAP Home Network Platform adds no Java interfaces and classes to packages defined in [OCAP].

### 6.7.2 OCAP Home Networking Packages

The following packages are defined by the OCAP Home Network Platform:

- org.ocap.hn
- org.ocap.hn.content
- org.ocap.hn.content.navigation
- org.ocap.hn.profiles.upnp
- org.ocap.hn.recording
- org.ocap.hn.service
- org.ocap.hn.security

These packages are described in Annex A through Annex G.

## 7 SECURITY

The OCAP Home Network Platform may not be permitted to share all digital entertainment Content that is stored on the OpenCable Host device or other Connected Devices. Content can only be shared according to CCI bits, as defined in [HOST]. OCAP Home Networking API methods, which cause Content to be streamed or moved to a remote device, SHALL fail if the request violates copy protection signaling, or other security requirements imposed by OpenCable security specifications. These security aspects are transparent to applications and the OCAP Home Networking API does not provide methods for application access and control of them.

### 7.1 Home Network Permission

In addition to device authorizations, this specification defines the `org.ocap.hn.HomeNetPermission` class. This class can be used to grant permissions for various OCAP Home Networking API methods. See the Annexes containing APIs for method details where `SecurityException` can be thrown. For granting purposes, the `HomeNetPermission` is added to the permission request file (PRF) of an application. Referring to the PRF DTD definition given in [OCAP], Section 14.2.2.1.1, this specification modifies the `permissionrequestfile` ELEMENT and adds the `ocap:homenetpermission` as follows:

```
<!ELEMENT permissionrequestfile
(file?, capermission?, applifecyclecontrol?, returnchannel?, tuning?,
servicesel?, userpreferences?, network?, dripfeed?, persistentfilecredential*,
ocap:monitorapplication*, ocap:serVICetypepermission*, ocap:homenetpermission*)>
```

The `ocap:homenetpermission` ELEMENT SHALL be added after the `ocap:serVICetypepermission` in the PRF as follows:

```
<!ELEMENT ocap:homenetpermission EMPTY>
<!ATTLIST ocap:homenetpermission
type (contentmanagement | contentlisting | recording | recordinghandler)
value (true | false) "false"
>
```

### 7.2 Monitor Application Permission

Host devices that implement the OCAP HN Extension SHALL support the `MonitorAppPermission("handler.homenetwork")` permission name. Annex Q of [OCAP] is extended as follows:

The table within the description of `MonitorAppPermission` is extended to include the following rows:

Permission Name	What the Permission Allows	Description
handler.homenetwork	Provides access to network passwords and authorization methods.	This permission allows the caller to get and set network passwords as well as register as the authorization handler.

In addition, Section 14.2.2.1.1 in [OCAP] is extended as follows: the enumerated token value type of the name attribute of the `ocap:monitorapplication` element type defined by the DTD of the PRF SHALL be considered to contain the "handler.homenetwork" values.

## 8 REGISTRY OF CONSTANTS

This section contains OCAP-specific constants for the home network profile. All constants are public final. These OCAP-specific JAVA constants are set in the following packages:

<b>org.ocap.hn.ContentServerEvent</b>		
public static final int	CONTENT_ADDED	0
public static final int	CONTENT_CHANGED	2
public static final int	CONTENT_REMOVED	1
<b>org.ocap.hn.Device</b>		
public static final java.lang.String	CAP_RECORDING_SUPPORTED	"RecordingSupported"
public static final java.lang.String	CAP_REMOTE_STORAGE_SUPPORTED	"RemoteStorageSupported"
public static final java.lang.String	CAP_STREAMING_SUPPORTED	"StreamingSupported"
public static final java.lang.String	CAP_TUNER_SUPPORTED	"TunerSupported"
public static final java.lang.String	PROP_DEVICE_TYPE	"deviceType"
public static final java.lang.String	PROP_DEVICE_VERSION	"deviceVersion"
public static final java.lang.String	PROP_FRIENDLY_NAME	"friendlyName"
public static final java.lang.String	PROP_LOCATION	"location"
public static final java.lang.String	PROP_MANUFACTURER	"manufacturer"
public static final java.lang.String	PROP_MANUFACTURER_URL	"manufacturerURL"
public static final java.lang.String	PROP_MIDDLEWARE_PROFILE	"middlewareProfile"
public static final java.lang.String	PROP_MIDDLEWARE_VERSION	"middlewareVersion"
public static final java.lang.String	PROP_MODEL_DESCRIPTION	"modelDescription"
public static final java.lang.String	PROP_MODEL_NAME	"modelName"
public static final java.lang.String	PROP_MODEL_NUMBER	"modelNameNumber"
public static final java.lang.String	PROP_MODEL_URL	"modelURL"
public static final java.lang.String	PROP_PRESENTATION_URL	"presentationURL"

<b>org.ocap.hn.Device</b>		
public static final java.lang.String	PROP_SERIAL_NUMBER	"serialNumber"
public static final java.lang.String	PROP_UDN	"UDN"
public static final java.lang.String	PROP_UPC	"UPC"
public static final java.lang.String	TYPE_BINARY_LIGHT	"BinaryLight"
public static final java.lang.String	TYPE_DIMMABLE_LIGHT	"DimmableLight"
public static final java.lang.String	TYPE_HVAC_SYSTEM	"HVAC_System"
public static final java.lang.String	TYPE_HVAC_ZONE_THERMOSTAT	"HVAC_ZoneThermostat"
public static final java.lang.String	TYPE_INTERNET_GATEWAY_DEVICE	"InternetGatewayDevice"
public static final java.lang.String	TYPE_LAN_DEVICE	"LANDevice"
public static final java.lang.String	TYPE_MEDIA_RENDERER	"MediaRenderer"
public static final java.lang.String	TYPE_MEDIA_SERVER	"MediaServer"
public static final java.lang.String	TYPE_OCAP_HOST	"OCAP_Host"
public static final java.lang.String	TYPE_OCAP_TERMINAL	"OCAP_Terminal"
public static final java.lang.String	TYPE_PRINTER	"printer"
public static final java.lang.String	TYPE_REMOTE_UI_CLIENT_DEVICE	"RemoteUIClientDevice"
public static final java.lang.String	TYPE_REMOTE_UI_SERVER_DEVICE	"RemoteUIServerDevice"
public static final java.lang.String	TYPE_SCANNER	"Scanner"
public static final java.lang.String	TYPE_WAN_CONNECTION_DEVICE	"WANConnectionDevice"
public static final java.lang.String	TYPE_WAN_DEVICE	"WANDevice"
public static final java.lang.String	TYPE_WLAN_ACCESS_POINT_DEVICE	"WLANAccessPointDevice"
<b>org.ocap.hn.DeviceEvent</b>		
public static final int	DEVICE_ADDED	100
public static final int	DEVICE_REMOVED	101
public static final int	DEVICE_UPDATED	102

<b>org.ocap.hn.DeviceEvent</b>		
public static final int	STATE_CHANGE	201
<b>org.ocap.hn.NetActionEvent</b>		
public static final int	ACTION_CANCELED	1
public static final int	ACTION_COMPLETED	0
public static final int	ACTION_FAILED	2
public static final int	ACTION_IN_PROGRESS	4
public static final int	ACTION_STATUS_NOT_AVAILABLE	3
<b>org.ocap.hn.NetModule</b>		
public static final java.lang.String	CONTENT_LIST	"ContentList"
public static final java.lang.String	CONTENT_MANAGER	"ContentManager"
public static final java.lang.String	CONTENT_RECORDER	"ContentRecorder"
public static final java.lang.String	CONTENT_RENDERER	"ContentRenderer"
public static final java.lang.String	CONTENT_SERVER	"ContentServer"
public static final java.lang.String	PROP_CONTROL_URL	"ControlURL"
public static final java.lang.String	PROP_DESCRIPTION_URL	"DescriptionURL"
public static final java.lang.String	PROP_EventSub_URL	"EventSubURL"
public static final java.lang.String	PROP_NETMODULE_ID	"NetModuleId"
public static final java.lang.String	PROP_NETMODULE_TYPE	"NetModuleType"
<b>org.ocap.hn.NetModuleEvent</b>		
public static final int	MODULE_ADDED	100
public static final int	MODULE_BUSY	103
public static final int	MODULE_REMOVED	101
public static final int	MODULE_UPDATED	102
public static final int	STATE_CHANGE	201
<b>org.ocap.hn.NetworkInterface</b>		
public static final int	MOCA	1
public static final int	UNKNOWN	0
public static final int	WIRED_ETHERNET	2
public static final int	WIRELESS_ETHERNET	3
<b>org.ocap.hn.content.ContentContainer</b>		
public static final java.lang.String	ALBUM_CONTAINER	"object.container.album"
public static final java.lang.String	ALBUM_CONTAINER_MUSIC	"object.container.album.musicAlbum"
public static final java.lang.String	ALBUM_CONTAINER_PHOTO	"object.container.album.photoAlbum"
public static final java.lang.String	CONTAINER	"object.container"

<b>org.ocap.hn.content.ContentContainer</b>		
public static final java.lang.String	GENRE_CONTAINER	"object.container.genre"
public static final java.lang.String	GENRE_CONTAINER_MOVIE	"object.container.genre .movieGenre"
public static final java.lang.String	GENRE_CONTAINER_MUSIC	"object.container.genre .musicGenre"
public static final java.lang.String	PERSON_CONTAINER	"object.container.person"
public static final java.lang.String	PERSON_CONTAINER_MUSIC _ARTIST	"object.container.person .musicArtist"
public static final java.lang.String	PLAYLIST_CONTAINER	"object.container .playlistContainer"
public static final java.lang.String	STORAGE_FOLDER_CONTAINER	"object.container.storageFolder"
public static final java.lang.String	STORAGE_SYSTEM_CONTAINER	"object.container.storageSystem"
public static final java.lang.String	STORAGE_VOLUME_CONTAINER	"object.container.storageVolume"
<b>org.ocap.hn.content.ContentItem</b>		
public static final java.lang.String	AUDIO_ITEM	"object.item.audioItem"
public static final java.lang.String	AUDIO_ITEM _BOOK	"object.item.audioItem.audioBook "
public static final java.lang.String	AUDIO_ITEM _BROADCAST	"object.item.audioItem.audioBroa dcast"
public static final java.lang.String	AUDIO_ITEM _TRACK	"object.item.audioItem.musicTrac k"
public static final java.lang.String	IMAGE_ITEM	"object.item.imageItem"
public static final java.lang.String	IMAGE_ITEM _PHOTO	"object.item.imageItem.photo"
public static final java.lang.String	ITEM	"object.item"
public static final java.lang.String	VIDEO_ITEM	"object.item.videoItem"
public static final java.lang.String	VIDEO_ITEM _BROADCAST	"object.item.videoItem.videoBroa dcast"
public static final java.lang.String	VIDEO_ITEM _MOVIE	"object.item.videoItem.movie"
public static final java.lang.String	VIDEO_ITEM _MUSIC_CLIP	"object.item.videoItem.musicVide oClip"
<b>org.ocap.hn.content.ContentResource</b>		
public static final java.lang.String	UNKNOWN_MIME_TYPE	"unknown"

<b>org.ocap.hn.content.DatabaseException</b>		
public static final int	FIELD_IS_EMPTY	3
public static final int	FIELD_IS_WRONG_FORMAT	4
public static final int	FIELD_NAME_DOES_NOT_EXIST	1
public static final int	GENERAL_ERROR	9
public static final int	INVALID_PARAMETER_SPECIFIED	6
public static final int	QUERY_IS_INVALID	5
public static final int	REMOTE_QUERY_IS_INVALID	8
public static final int	UNABLE_TO_LOCATE_SERVICE	7
<b>org.ocap.hn.content.MetadataIdentifiers</b>		
public static final java.lang.String	PROPRIETARY_DATA	"ocap:proprietaryData"
<b>org.ocap.hn.content.navigation.DatabaseQuery</b>		
public static final int	CONTAINS	6
public static final int	EQUALS	1
public static final int	EXISTS	8
public static final int	GREATER_THAN	2
public static final int	GREATER_THAN_OR_EQUALS	4
public static final int	LESS_THAN	3
public static final int	LESS_THAN_OR_EQUALS	5
public static final int	NOT_EQUALS	7
<b>org.ocap.hn.profiles.upnp.UPnPConstants</b>		
public static final java.lang.String	ACTOR	"upnp:actor"
public static final java.lang.String	ACTOR_AT_ROLE	"upnp:actor@role"
public static final java.lang.String	ALBUM	"upnp:album"
public static final java.lang.String	ALBUM_ART	"upnp:albumArtURI"
public static final java.lang.String	ARTIST	"upnp:artist"
public static final java.lang.String	ARTIST_AT_ROLE	"upnp:artist@role"
public static final java.lang.String	ARTIST_DISCOGRAPHY	"upnp:albumArtURI"
public static final java.lang.String	AUTHOR	"upnp:author"
public static final java.lang.String	AUTHOR_AT_ROLE	"upnp:actor@role"
public static final java.lang.String	CHANNEL_NAME	"upnp:channelName"

<b>org.ocap.hn.profiles.upnp.UPnPConstants</b>		
public static final java.lang.String	CHANNEL_NUMBER	"upnp:channelNr"
public static final java.lang.String	COMMENTS	"upnp:userAnnotation"
public static final java.lang.String	CONTRIBUTOR	"dc:contributor"
public static final java.lang.String	CREATION_DATE	"dc:date"
public static final java.lang.String	CREATOR	"dc:creator"
public static final java.lang.String	DESCRIPTION	"dc:description"
public static final java.lang.String	DIRECTOR	"upnp:director"
public static final java.lang.String	DVD_REGION_CODE	"upnp:DVDRegionCode"
public static final java.lang.String	GENRE	"upnp:genre"
public static final java.lang.String	ICON_REF	"upnp:icon"
public static final java.lang.String	ID	"id"
public static final java.lang.String	LANGUAGE	"dc:language"
public static final java.lang.String	LONG_DESCRIPTION	"upnp:longDescription"
public static final java.lang.String	LYRICS_REF	"upnp:lyricsURI"
public static final java.lang.String	MEDIA_ID	"upnp:toc"
public static final java.lang.String	PARENT_ID	"parentID"
public static final java.lang.String	PLAYLIST	"upnp:playlist"
public static final java.lang.String	PRODUCER	"upnp:producer"
public static final java.lang.String	PROP_STORAGE_FREE	"upnp:storageFree"
public static final java.lang.String	PROP_STORAGE_TOTAL	"upnp:storageTotal"
public static final java.lang.String	PUBLISHER	"dc:publisher"
public static final java.lang.String	RADIO_BAND	"upnp:radioBand"
public static final	RADIO_CALL_SIGN	"upnp:radioCallSign"

<b>org.ocap.hn.profiles.upnp.UPnPConstants</b>		
java.lang.String		
public static final java.lang.String	RADIO_STATION_ID	"upnp:radioStationID"
public static final java.lang.String	RATING	"upnp:rating"
public static final java.lang.String	REGION	"upnp:region"
public static final java.lang.String	RELATION	"dc:relation"
public static final java.lang.String	RIGHTS	"dc:rights"
public static final java.lang.String	SCHEDULED_END_TIME	"upnp:scheduledEndTime"
public static final java.lang.String	SCHEDULED_START_TIME	"upnp:scheduledStartTime"
public static final java.lang.String	STORAGE_MEDIUM	"upnp:storageMedium"
public static final java.lang.String	TITLE	"dc:title"
public static final java.lang.String	TRACK_NUMBER	"upnp:originalTrackNumber"
<b>org.ocap.hn.recording.NetRecordingEntry</b>		
public static final java.lang.String	PROP_NET_RECORDING_ENTRY	"ocap:isNetRecordingEntry"
<b>org.ocap.hn.recording.RecordingContentItem</b>		
public static final java.lang.String	PROP_ACCESS_PERMISSIONS	"ocap:accessPermissions"
public static final java.lang.String	PROP_APP_ID	"ocap:appID"
public static final java.lang.String	PROP_CONTENT_URI	"ocap:contentURI"
public static final java.lang.String	PROP_DELETION_REASON	"ocap:deletionReason"
public static final java.lang.String	PROP_DELETION_TIME	"ocap:deletionTime"
public static final java.lang.String	PROP_DESTINATION	"ocap:destination"
public static final java.lang.String	PROP_DURATION	"ocap:scheduledDuration"
public static final java.lang.String	PROP_EXPIRATION_PERIOD	"ocap:expirationPeriod"
public static final java.lang.String	PROP_FAILURE_REASON	"ocap:failureReason"
public static final	PROP_MSO_CONTENT	"ocap:msoContentIndicator"

<b>org.ocap.hn.recording.RecordingContentItem</b>		
java.lang.String		
public static final java.lang.String	PROP_NET_RECORDING_ENTRY	"ocap:netRecordingEntry"
public static final java.lang.String	PROP_ORGANIZATION	"ocap:organization"
public static final java.lang.String	PROP_PRESENTATION_POINT	"ocap:mediaPresentationPoint"
public static final java.lang.String	PROP_PRIORITY_FLAG	"ocap:priorityFlag"
public static final java.lang.String	PROP_RECORDING_STATE	"ocap:taskState"
public static final java.lang.String	PROP_RETENTION_PRIORITY	"ocap:retentionPriority"
public static final java.lang.String	PROP_SOURCE_ID	"ocap:scheduledChannelID"
public static final java.lang.String	PROP_SPACE_REQUIRED	"ocap:spaceRequired"
public static final java.lang.String	PROP_START_TIME	"ocap:scheduledStartDateTime"

## Annex A Home Networking API

Package org.ocap.hn

Interface Summary	
<b>ContentServerListener</b>	Listener interface for classes which are interested in changes to a ContentServerNetModule.
<b>ContentServerNetModule</b>	Class representing a NetModule which serves content.
<b>Device</b>	The Device interface represents a Homenetwork device that supports homenetWORK NetModules.
<b>DeviceEventListener</b>	DeviceEvent callback interface.
<b>NetActionHandler</b>	This interface represents a handler passed to asynchronous methods
<b>NetActionRequest</b>	All asynchronous actions in the Home networking API return an NetActionRequest.
<b>NetList</b>	A list comprising of homenetWORK elements such as Device or NetModule.
<b>NetModule</b>	NetModule is an abstraction of functionality that is provided by a Device.
<b>NetModuleEventListener</b>	NetModuleEvent callback interface.

Class Summary	
<b>ContentServerEvent</b>	Event which will be sent to registered ContentServerListeners when ContentEntrys have been added, changed or removed.
<b>DeviceEvent</b>	Represents a Device Event.
<b>HomeNetPermission</b>	The HomeNetPermission class represents permission to execute privileged home networking operations only signed applications MAY be granted.
<b>NetActionEvent</b>	This class represents an event generated in response to an action.
<b>NetManager</b>	The NetManager is a singleton class that registers all the Devices and NetModules within a home network.
<b>NetModuleEvent</b>	Entity for NetModule Event.
<b>NetworkInterface</b>	This class represents a home network interface including MoCA, wired ethernet, and wireless ethernet.
<b>PropertyFilter</b>	The filter for (key,value) pair filtering mechanism.

Exception Summary	
<b>NotAuthorizedException</b>	Exception indicating that the application has no permission to perform certain action.

org.ocap.hn

**Class ContentServerEvent**

```

java.lang.Object
├─ java.util.EventObject
│   └─ org.ocap.hn.ContentServerEvent

```

**All Implemented Interfaces:**

java.io.Serializable

```

public class ContentServerEvent
extends java.util.EventObject

```

Event which will be sent to registered ContentServerListeners when ContentEntrys have been added, changed or removed.

**See Also:**

Serialized Form

**Field Summary**

static int	<b>CONTENT_ADDED</b> Event ID indicating that content got added to a ContentServerNetModule.
static int	<b>CONTENT_CHANGED</b> Event ID indicating that metadata associated with content has been updated.
static int	<b>CONTENT_REMOVED</b> Event ID indicating that content got removed from a ContentServerNetModule

**Fields inherited from class java.util.EventObject**

source

**Constructor Summary**

**ContentServerEvent**(java.lang.Object source, java.lang.String[] content, int evt)

Creates a new ContentServerEvent with the given source object, the ContentItem involved and an event ID indicating whether the content got added or removed.

**Method Summary**

java.lang.String[]	<b>getContent</b> () Returns the IDs associated with the ContentEntrys involved in this event.
ContentServerNetModule	<b>getContentServerNetModule</b> () Returns the ContentServerNetModule.
int	<b>getEventID</b> () Gets the event ID for this event.

**Methods inherited from class java.util.EventObject**

getSource, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

**CONTENT\_ADDED**

```
public static final int CONTENT_ADDED
```

Event ID indicating that content got added to a ContentServerNetModule. This event SHALL NOT guarantee that any content items or associated metadata have been communicated to the local device. Applications should utilize the content browsing and searching APIs to retrieve any added content items.

**See Also:**

ContentServerNetModule, Constant Field Values

**CONTENT\_REMOVED**

```
public static final int CONTENT_REMOVED
```

Event ID indicating that content got removed from a ContentServerNetModule

**See Also:**

Constant Field Values

**CONTENT\_CHANGED**

```
public static final int CONTENT_CHANGED
```

Event ID indicating that metadata associated with content has been updated. This event SHALL NOT guarantee that and changes to content items or associated metadata have been communicated to the local device. Applications should utilize the content browsing and searching APIs to retrieve any updated metadata.

**See Also:**

ContentServerNetModule, Constant Field Values

## Constructor Detail

**ContentServerEvent**

```
public ContentServerEvent(java.lang.Object source,
                          java.lang.String[] content,
                          int evt)
```

Creates a new ContentServerEvent with the given source object, the ContentItem involved and an event ID indicating whether the content got added or removed.

**Parameters:**

source - The source of this event. This must be a ContentServerNetModule.

content - the IDs of the ContentEntries involved.

evt - the Event ID, either CONTENT\_ADDED, CONTENT\_REMOVED or CONTENT\_CHANGED.

## Method Detail

### **getContent**

```
public java.lang.String[] getContent()
```

Returns the IDs associated with the ContentEntrys involved in this event.

**Returns:**

the string IDs of the entries involved.

### **getContentServerNetModule**

```
public ContentServerNetModule getContentServerNetModule()
```

Returns the ContentServerNetModule. This is the source object of the event.

**Returns:**

the ContentServerNetModule containing the ContentItem that was added/removed/changed

### **getEventID**

```
public int getEventID()
```

Gets the event ID for this event. Valid values are CONTENT\_ADDED, CONTENT\_CHANGED and CONTENT\_REMOVED

**Returns:**

the ID for this event

org.ocap.hn

## Interface ContentServerListener

### All Superinterfaces:

java.util.EventListener

```
public interface ContentServerListener
extends java.util.EventListener
```

Listener interface for classes which are interested in changes to a ContentServerNetModule.

## Method Summary

void	<b>contentUpdated</b> (ContentServerEvent evt) Called when a ContentEntry has been added, changed or removed from the ContentServerNetModule
------	---

## Method Detail

### contentUpdated

```
void contentUpdated(ContentServerEvent evt)
    Called when a ContentEntry has been added, changed or removed from the
    ContentServerNetModule
```

#### Parameters:

evt - the ContentServerEvent

**org.ocap.hn****Interface ContentServerNetModule****All Superinterfaces:**

NetModule

```
public interface ContentServerNetModule
extends NetModule
```

Class representing a NetModule which serves content.

NetModules which implement this interface SHALL have a NetModule.PROP\_NETMODULE\_TYPE property value of NetModule.CONTENT\_SERVER.

## Field Summary

### Fields inherited from interface org.ocap.hn.NetModule

CONTENT\_LIST, CONTENT\_MANAGER, CONTENT\_RECORDER, CONTENT\_RENDERER,  
CONTENT\_SERVER, PROP\_CONTROL\_URL, PROP\_DESCRIPTION\_URL, PROP\_EventSub\_URL,  
PROP\_NETMODULE\_ID, PROP\_NETMODULE\_TYPE

## Method Summary

void	<b>addContentServerListener</b> (ContentServerListener listener) Adds a ContentServerListener to this ContentContainer.
void	<b>removeContentServerListener</b> (ContentServerListener listener) Removes the specified ContentServerListener.
NetActionRequest	<b>requestBrowseEntries</b> (java.lang.String startingEntryID, java.lang.String propertyFilter, boolean browseChildren, int startingIndex, int requestedCount, java.lang.String sortCriteria, NetActionHandler handler) Requests a browse of this ContentServer which results in the creation of a ContentList.
NetActionRequest	<b>requestRootContainer</b> (NetActionHandler handler) returns the root ContentContainer for this ContentServerNetModule.
NetActionRequest	<b>requestSearchCapabilities</b> (NetActionHandler handler) Returns the list of property keys which applications can search against on this ContentServer using the requestSearchEntries(java.lang.String, java.lang.String, int, int, java.lang.String, java.lang.String, org.ocap.hn.NetActionHandler) method.
NetActionRequest	<b>requestSearchEntries</b> (java.lang.String parentID, java.lang.String propertyFilter, int startingIndex, int requestedCount, java.lang.String searchCriteria, java.lang.String sortCriteria, NetActionHandler handler) Requests a search of this ContentServer which results in the creation of a ContentList.

**Methods inherited from interface org.ocap.hn.NetModule**

addNetModuleEventListener, getDevice, getKeys, getNetModuleId, getNetModuleType, getProperty, isLocal, removeNetModuleEventListener

**Method Detail****requestRootContainer**

NetActionRequest **requestRootContainer**(NetActionHandler handler)

returns the root ContentContainer for this ContentServerNetModule. This is an asynchronous method. The caller gets informed via NetActionHandler.notify(NetActionEvent) of the process. On success, a NetActionEvent is created where the NetActionEvent.getResponse() method will return a ContentContainer object representing the root container for this ContentServerNetModule.

**Parameters:**

handler - NetActionHandler which gets informed once this asynchronous request completes

**Returns:**

NetActionRequest See NetActionRequest

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("contentlisting")

**requestSearchCapabilities**

NetActionRequest **requestSearchCapabilities**(NetActionHandler handler)

Returns the list of property keys which applications can search against on this ContentServer using the requestSearchEntries(java.lang.String, java.lang.String, int, int, java.lang.String, java.lang.String, org.ocap.hn.NetActionHandler) method. This is an asynchronous method. The caller gets informed via NetActionHandler.notify(NetActionEvent) of the process. On success an NetActionEvent is created where the NetActionEvent.getResponse() method will return an array of String objects containing the valid property keys. A return of an array with zero length indicates that this server supports no searching functionality. A return containing "\*" indicates that any key associated with any content entry on this server may be used.

**Parameters:**

handler - NetActionHandler which gets informed once this asynchronous request completes

**Returns:**

NetActionRequest See NetActionRequest

**requestBrowseEntries**

NetActionRequest **requestBrowseEntries**(java.lang.String startingEntryID, java.lang.String propertyFilter, boolean browseChildren, int startingIndex, int requestedCount, java.lang.String sortCriteria, NetActionHandler handler)

Requests a browse of this ContentServer which results in the creation of a ContentList.

ContentEntry objects hosted on the remote server will be browsed starting at the ContentEntry specified. The propertyFilter parameter of this method SHALL contain a comma separated list of properties indicating which metadata fields should be returned in the ContentEntry objects contained in the resulting ContentList. A filter value of "\*" indicates all available metadata be returned. The sortCriteria parameter of this method is a string containing the properties and sort modifiers to be used to sort the

resulting `ContentList`. The format of the string containing the sort criteria shall follow the format defined in UPnP Content Directory Service 3.0 specification section 2.3.16: `A_ARG_TYPE_SortCriteria`.

This is an asynchronous method. The caller gets informed via `NetActionHandler.notify(NetActionEvent)` of the process. On success an `NetActionEvent` is created where the `NetActionEvent.getResponse()` method will return a `ContentList` containing the search results. If no matches are found, this value SHALL be a `ContentList` with zero entries. A return from `NetActionEvent.getActionStatus()` of `NetActionEvent.ACTION_COMPLETED` SHALL indicate that a valid `ContentList` will be returned from `NetActionEvent.getResponse()`.

**Parameters:**

`startingEntryID` - the ID of the `ContentEntry` on the server to start the browse from. A value of "0" SHALL indicate the root container on this server.

`propertyFilter` - the set of property values to return from this browse operation

`browseChildren` - if set to true, this operation will browse all of the direct children of the `startingEntryID` parameter. If false, this operation will return a content list containing the entry identified by `startingEntryID` only.

`startingIndex` - starting zero-based offset to enumerate children under the container specified by parent.

`requestedCount` - requested number of entries under the `ContentContainer` specified by parent. Setting this parameter to 0 indicates request all entries.

`sortCriteria` - properties and sort modifiers to be used to sort the resulting `ContentList`

`handler` - `NetActionHandler` which gets informed once the results `ContentList` is created or an error occurs. calling `getResponse()` on handler will return a `ContentList` containing the requested entries, or if the call was unsuccessful will return an error message supplied by the server.

**Returns:**

`NetActionRequest` See `NetActionRequest`.

**Throws:**

`java.lang.IllegalArgumentException` - if the `startingEntryID` is not available on this `ContentServerNetModule`, or if the handler parameter is null.

`java.lang.SecurityException` - if the caller does not have `HomeNetPermission("contentlisting")`

## requestSearchEntries

```
NetActionRequest requestSearchEntries( java.lang.String parentID,
                                       java.lang.String propertyFilter,
                                       int startingIndex,
                                       int requestedCount,
                                       java.lang.String searchCriteria,
                                       java.lang.String sortCriteria,
                                       NetActionHandler handler)
```

Requests a search of this `ContentServer` which results in the creation of a `ContentList`.

`ContentEntry` objects hosted on the remote server will be searched for using the specified search criteria. The format of the string containing the search criteria SHALL follow the format defined by the UPnP Content Directory Service 3.0 specification section 2.3.13.1: Search Criteria String Syntax. The `propertyFilter` parameter of this method SHALL contain a comma separated list of properties indicating which metadata fields should be returned in the `ContentEntry` objects contained in the resulting `ContentList`. A filter value of "\*" indicates all available metadata be returned. The `sortCriteria` parameter of this method is a string containing the properties and sort modifiers to be used to sort the resulting `ContentList`. The format of the string containing the sort criteria shall follow the format defined in UPnP Content Directory Service 3.0 specification section 2.3.16: `A_ARG_TYPE_SortCriteria`.

This is an asynchronous method. The caller gets informed via `NetActionHandler.notify(NetActionEvent)` of the process. On success an

`NetActionEvent` is created where the `NetActionEvent.getResponse()` method will return a `ContentList` containing the search results. If no matches are found, this value SHALL be a `ContentList` with zero entries. A return from `NetActionEvent.getActionStatus()` of `NetActionEvent.ACTION_COMPLETED` SHALL indicate that a valid `ContentList` will be returned from `NetActionEvent.getResponse()`.

**Parameters:**

`parentID` - the ID of the `ContentContainer` on the server to start the search from. A value of "0" SHALL indicate the root container on this server.

`propertyFilter` - the set of property values to return from this browse operation

`startingIndex` - starting zero-based offset to enumerate children under the container specified by `parent`.

`requestedCount` - requested number of entries under the `ContentContainer` specified by `parent`. Setting this parameter to 0 indicates request all entries.

`searchCriteria` - contains the criteria string to search for. If this parameter is null, the implementation SHALL consider all entries in the parent container as matching the search criteria.

`sortCriteria` - properties and sort modifiers to be used to sort the resulting `ContentList`

`handler` - `NetActionHandler` which gets informed once the results `ContentList` is created or an error occurs. calling `getResponse()` on `handler` will return a `ContentList` containing the requested entries, or if the call was unsuccessful will return an error message supplied by the server.

**Returns:**

`NetActionRequest` See `NetActionRequest`.

**Throws:**

`java.lang.IllegalArgumentException` - if the `startingEntryID` is not available on this `ContentServerNetModule`, or if the `handler` parameter is null.

`java.lang.SecurityException` - if the caller does not have `HomeNetPermission("contentlisting")`

**addContentServerListener**

```
void addContentServerListener(ContentServerListener listener)
```

Adds a `ContentServerListener` to this `ContentContainer`. This `ContentServerListener` will be notified of additions, removals, or changes to any objects contained within this server

**Parameters:**

`listener` - the `Listener` that will receive `ContentServerEvents`.

**removeContentServerListener**

```
void removeContentServerListener(ContentServerListener listener)
```

Removes the specified `ContentServerListener`.

**Parameters:**

`listener` - the `Listener` to remove

## org.ocap.hn Interface Device

public interface **Device**

The Device interface represents a Homenetwork device that supports homenetwork NetModules. A Device is a hierarchical structure with root device being the physical appliance, such as an OCAP\_Terminal or an OCAP\_HOST. The valid device types for an OCAP root device are OCAP\_HOST and OCAP\_Terminal. A root device may contain a number of sub-devices, such as a MediaServer or a MediaRenderer. Each sub-device may support one or more NetModule(s) whereas each NetModule only represents one sub-device. A NetModule is some functional unit in the device and examples of NetModules are ContentList, ContentManager, etc. A device may also have certain capabilities and properties associated with it. An application can retrieve these capabilities and properties by using property filters

Field Summary	
static java.lang.String	<b>CAP_RECORDING_SUPPORTED</b> A constant indicating MSO content recording capability.
static java.lang.String	<b>CAP_REMOTE_STORAGE_SUPPORTED</b> A constant indicating remote storage capability.
static java.lang.String	<b>CAP_STREAMING_SUPPORTED</b> A constant indicating streaming capability of the device.
static java.lang.String	<b>CAP_TUNER_SUPPORTED</b> A constant indicating if the device has a tuner.
static java.lang.String	<b>PROP_DEVICE_TYPE</b> A constant indicates device property: device type
static java.lang.String	<b>PROP_DEVICE_VERSION</b> A constant representing a device version number
static java.lang.String	<b>PROP_FRIENDLY_NAME</b> A constant for a friendly name of the device.
static java.lang.String	<b>PROP_LOCATION</b> A constant indicates device property: location of the device.
static java.lang.String	<b>PROP_MANUFACTURER</b> A constant indicating the manufacturer of this device.
static java.lang.String	<b>PROP_MANUFACTURER_URL</b> A constant providing URL to the manufacturer's web site.
static java.lang.String	<b>PROP_MIDDLEWARE_PROFILE</b> A constant indicates device property: middleware profile.
static java.lang.String	<b>PROP_MIDDLEWARE_VERSION</b> A constant indicates device property: middleware version.
static java.lang.String	<b>PROP_MODEL_DESCRIPTION</b> A constant providing description of the device.
static java.lang.String	<b>PROP_MODEL_NAME</b> A constant indicates device property: model name.

<b>Field Summary</b>	
static java.lang.String	<b>PROP_MODEL_NUMBER</b> A constant indicates device property: model number.
static java.lang.String	<b>PROP_MODEL_URL</b> A constant indicates device property: model URL.
static java.lang.String	<b>PROP_PRESENTATION_URL</b> A constant indicates device property: presentation URL.
static java.lang.String	<b>PROP_SERIAL_NUMBER</b> A constant indicates device property: serial number.
static java.lang.String	<b>PROP_UDN</b> A constant indicates device property: unique device name.
static java.lang.String	<b>PROP_UPC</b> A constant indicates device property: universal product code.
static java.lang.String	<b>TYPE_BINARY_LIGHT</b> A constant indicates device type: Binary Light (on/off).
static java.lang.String	<b>TYPE_DIMMABLE_LIGHT</b> A constant indicates device type: Dimmable Light (light intensity control).
static java.lang.String	<b>TYPE_HVAC_SYSTEM</b> A constant indicates device type: Heater-Vent-Air Conditioning System.
static java.lang.String	<b>TYPE_HVAC_ZONE_THERMOSTAT</b> A constant indicates device type: Heater-Vent-Air Conditioning Thermostat.
static java.lang.String	<b>TYPE_INTERNET_GATEWAY_DEVICE</b> A constant indicates device type: Internet gateway device.
static java.lang.String	<b>TYPE_LAN_DEVICE</b> A constant indicates device type: LAN device.
static java.lang.String	<b>TYPE_MEDIA_RENDERER</b> A constant indicates device type: Media Renderer.
static java.lang.String	<b>TYPE_MEDIA_SERVER</b> A constant indicates device type: Media Server.
static java.lang.String	<b>TYPE_OCAP_HOST</b> A constant indicates device type: OCAP Host.
static java.lang.String	<b>TYPE_OCAP_TERMINAL</b> A constant indicates device type: OCAP terminal.
static java.lang.String	<b>TYPE_PRINTER</b> A constant indicates device type: Printer.
static java.lang.String	<b>TYPE_REMOTE_UI_CLIENT_DEVICE</b> A constant indicates device type: Remote UI Client Device, Allows for basic operations on a Remote UI client including: user interface connection management, optionally user interface availability management and optionally basic user interaction.
static java.lang.String	<b>TYPE_REMOTE_UI_SERVER_DEVICE</b> A constant indicates device type: Remote UI Server Device.

Field Summary	
static java.lang.String	<b>TYPE_SCANNER</b> A constant indicates device type: Scanner.
static java.lang.String	<b>TYPE_WAN_CONNECTION_DEVICE</b> A constant indicates device type: WAN connection device.
static java.lang.String	<b>TYPE_WAN_DEVICE</b> A constant indicates device type: WAN device.
static java.lang.String	<b>TYPE_WLAN_ACCESS_POINT_DEVICE</b> A constant indicates device type: WAN access point device.

Method Summary	
void	<b>addDeviceEventListener</b> (DeviceEventListener listener) Adds a DeviceEventListener instance to this Device.
java.util.Enumeration	<b>getCapabilities</b> () Returns capabilities of this device in Enumeration.
java.net.InetAddress	<b>getInetAddress</b> () Returns the IP address for this device.
java.util.Enumeration	<b>getKeys</b> () Returns all property keys supported by this device in Enumeration.
java.lang.String	<b>getName</b> () Returns the name of this device.
NetModule	<b>getNetModule</b> (java.lang.String moduleId) Returns the NetModule by module id.
NetList	<b>getNetModuleList</b> () Returns the list of NetModules supported by this device.
Device	<b>getParentDevice</b> () Returns the parent of this device.
java.lang.String	<b>getProperty</b> (java.lang.String key) Returns property of this device specified by a key.
NetList	<b>getSubDevices</b> () Returns a list of sub devices hosted by this device.
java.lang.String	<b>getType</b> () Returns the type of this device, for example, MediaRenderer, MediaServer, etc.
java.lang.String	<b>getVersion</b> () Returns the version number associated with this Device's device type.
boolean	<b>isLocal</b> () Returns true when this is the local device.
void	<b>removeDeviceEventListener</b> (DeviceEventListener listener) Removes a DeviceEventListener instance from this Device.
void	<b>setFriendlyName</b> (java.lang.String value) Sets the value of the PROP_FRIENDLY_NAME property.

## Field Detail

### **CAP\_STREAMING\_SUPPORTED**

static final java.lang.String **CAP\_STREAMING\_SUPPORTED**

A constant indicating streaming capability of the device.

**See Also:**

Constant Field Values

### **CAP\_TUNER\_SUPPORTED**

static final java.lang.String **CAP\_TUNER\_SUPPORTED**

A constant indicating if the device has a tuner.

**See Also:**

Constant Field Values

### **CAP\_REMOTE\_STORAGE\_SUPPORTED**

static final java.lang.String **CAP\_REMOTE\_STORAGE\_SUPPORTED**

A constant indicating remote storage capability.

**See Also:**

Constant Field Values

### **CAP\_RECORDING\_SUPPORTED**

static final java.lang.String **CAP\_RECORDING\_SUPPORTED**

A constant indicating MSO content recording capability.

**See Also:**

Constant Field Values

### **PROP\_FRIENDLY\_NAME**

static final java.lang.String **PROP\_FRIENDLY\_NAME**

A constant for a friendly name of the device.

**See Also:**

Constant Field Values

### **PROP\_MANUFACTURER**

static final java.lang.String **PROP\_MANUFACTURER**

A constant indicating the manufacturer of this device.

**See Also:**

Constant Field Values

### **PROP\_MANUFACTURER\_URL**

static final java.lang.String **PROP\_MANUFACTURER\_URL**

A constant providing URL to the manufacturer's web site.

**See Also:**

Constant Field Values

### **PROP\_MODEL\_DESCRIPTION**

static final java.lang.String **PROP\_MODEL\_DESCRIPTION**

A constant providing description of the device.

**See Also:**

Constant Field Values

### **PROP\_MODEL\_NAME**

`static final java.lang.String PROP_MODEL_NAME`

A constant indicates device property: model name.

**See Also:**

Constant Field Values

### **PROP\_MODEL\_NUMBER**

`static final java.lang.String PROP_MODEL_NUMBER`

A constant indicates device property: model number.

**See Also:**

Constant Field Values

### **PROP\_MODEL\_URL**

`static final java.lang.String PROP_MODEL_URL`

A constant indicates device property: model URL.

**See Also:**

Constant Field Values

### **PROP\_SERIAL\_NUMBER**

`static final java.lang.String PROP_SERIAL_NUMBER`

A constant indicates device property: serial number.

**See Also:**

Constant Field Values

### **PROP\_UDN**

`static final java.lang.String PROP_UDN`

A constant indicates device property: unique device name.

**See Also:**

Constant Field Values

### **PROP\_UPC**

`static final java.lang.String PROP_UPC`

A constant indicates device property: universal product code.

**See Also:**

Constant Field Values

### **PROP\_PRESENTATION\_URL**

`static final java.lang.String PROP_PRESENTATION_URL`

A constant indicates device property: presentation URL.

**See Also:**

Constant Field Values

**PROP\_LOCATION**

```
static final java.lang.String PROP_LOCATION
```

A constant indicates device property: location of the device.  
**See Also:**  
Constant Field Values

**PROP\_MIDDLEWARE\_PROFILE**

```
static final java.lang.String PROP_MIDDLEWARE_PROFILE
```

A constant indicates device property: middleware profile.  
**See Also:**  
Constant Field Values

**PROP\_MIDDLEWARE\_VERSION**

```
static final java.lang.String PROP_MIDDLEWARE_VERSION
```

A constant indicates device property: middleware version.  
**See Also:**  
Constant Field Values

**PROP\_DEVICE\_TYPE**

```
static final java.lang.String PROP_DEVICE_TYPE
```

A constant indicates device property: device type  
**See Also:**  
Constant Field Values

**PROP\_DEVICE\_VERSION**

```
static final java.lang.String PROP_DEVICE_VERSION
```

A constant representing a device version number  
**See Also:**  
Constant Field Values

**TYPE\_HVAC\_SYSTEM**

```
static final java.lang.String TYPE_HVAC_SYSTEM
```

A constant indicates device type: Heater-Vent-Air Conditioning System.  
**See Also:**  
Constant Field Values

**TYPE\_HVAC\_ZONE\_THERMOSTAT**

```
static final java.lang.String TYPE_HVAC_ZONE_THERMOSTAT
```

A constant indicates device type: Heater-Vent-Air Conditioning Thermostat.  
**See Also:**  
Constant Field Values

**TYPE\_INTERNET\_GATEWAY\_DEVICE**

```
static final java.lang.String TYPE_INTERNET_GATEWAY_DEVICE
```

A constant indicates device type: Internet gateway device.  
**See Also:**  
Constant Field Values

**TYPE\_LAN\_DEVICE**

static final java.lang.String **TYPE\_LAN\_DEVICE**  
A constant indicates device type: LAN device.  
**See Also:**  
Constant Field Values

**TYPE\_WAN\_CONNECTION\_DEVICE**

static final java.lang.String **TYPE\_WAN\_CONNECTION\_DEVICE**  
A constant indicates device type: WAN connection device.  
**See Also:**  
Constant Field Values

**TYPE\_WAN\_DEVICE**

static final java.lang.String **TYPE\_WAN\_DEVICE**  
A constant indicates device type: WAN device.  
**See Also:**  
Constant Field Values

**TYPE\_BINARY\_LIGHT**

static final java.lang.String **TYPE\_BINARY\_LIGHT**  
A constant indicates device type: Binary Light (on/off).  
**See Also:**  
Constant Field Values

**TYPE\_DIMMABLE\_LIGHT**

static final java.lang.String **TYPE\_DIMMABLE\_LIGHT**  
A constant indicates device type: Dimmable Light (light intensity control).  
**See Also:**  
Constant Field Values

**TYPE\_MEDIA\_SERVER**

static final java.lang.String **TYPE\_MEDIA\_SERVER**  
A constant indicates device type: Media Server.  
**See Also:**  
Constant Field Values

**TYPE\_MEDIA\_RENDERER**

static final java.lang.String **TYPE\_MEDIA\_RENDERER**  
A constant indicates device type: Media Renderer.  
**See Also:**  
Constant Field Values

**TYPE\_PRINTER**

static final java.lang.String **TYPE\_PRINTER**  
A constant indicates device type: Printer.  
**See Also:**

## Constant Field Values

**TYPE\_REMOTE\_UI\_CLIENT\_DEVICE**

```
static final java.lang.String TYPE_REMOTE_UI_CLIENT_DEVICE
```

A constant indicates device type: Remote UI Client Device, Allows for basic operations on a Remote UI client including: user interface connection management, optionally user interface availability management and optionally basic user interaction.

**See Also:**

Constant Field Values

**TYPE\_REMOTE\_UI\_SERVER\_DEVICE**

```
static final java.lang.String TYPE_REMOTE_UI_SERVER_DEVICE
```

A constant indicates device type: Remote UI Server Device.

**See Also:**

TYPE\_REMOTE\_UI\_CLIENT\_DEVICE, Constant Field Values

**TYPE\_SCANNER**

```
static final java.lang.String TYPE_SCANNER
```

A constant indicates device type: Scanner.

**See Also:**

Constant Field Values

**TYPE\_WLAN\_ACCESS\_POINT\_DEVICE**

```
static final java.lang.String TYPE_WLAN_ACCESS_POINT_DEVICE
```

A constant indicates device type: WAN access point device.

**See Also:**

Constant Field Values

**TYPE\_OCAP\_HOST**

```
static final java.lang.String TYPE_OCAP_HOST
```

A constant indicates device type: OCAP Host.

**See Also:**

Constant Field Values

**TYPE\_OCAP\_TERMINAL**

```
static final java.lang.String TYPE_OCAP_TERMINAL
```

A constant indicates device type: OCAP terminal.

**See Also:**

Constant Field Values

## Method Detail

**getCapabilities**

```
java.util.Enumeration getCapabilities()
```

Returns capabilities of this device in Enumeration. Capabilities are defined in Device.

**Returns:**

An enumeration of String objects representing capabilities of this device.

**getName**

```
java.lang.String getName()
```

Returns the name of this device. Device name is unique within a home network. The naming rule is proprietary. For example, "LivingRoom:OCAP\_HOST1".

**Returns:**

name of this device

**getProperty**

```
java.lang.String getProperty(java.lang.String key)
```

Returns property of this device specified by a key. Minimum supported keys are defined in Device, like PROP\_MANUFACTURER, PROP\_MODEL\_NUMBER, etc.

**Parameters:**

key - key of the property

**Returns:**

property value specified by the key

**getKeys**

```
java.util.Enumeration getKeys()
```

Returns all property keys supported by this device in Enumeration. Keys returned may include standardized keys (as documented with constants in this interface), as well as additional keys supported by this device.

**Returns:**

An enumeration of String objects representing all property keys supported by this device

**getNetModuleList**

```
NetList getNetModuleList()
```

Returns the list of NetModules supported by this device.

**Returns:**

NetList supported by this device

**getNetModule**

```
NetModule getNetModule(java.lang.String moduleId)
```

Returns the NetModule by module id. Module id is unique within a device.

**Parameters:**

moduleId - unique id of a NetModule

**Returns:**

NetModule by id, if specified NetModule is not supported by this device, then null is returned.

**getSubDevices**

```
NetList getSubDevices()
```

Returns a list of sub devices hosted by this device.

**Returns:**

list of sub-devices.

**getParentDevice**

```
Device getParentDevice()
```

Returns the parent of this device.

**Returns:**  
the parent device, or null if this device has no parent.

### getType

java.lang.String **getType()**  
Returns the type of this device, for example, MediaRenderer, MediaServer, etc. All OCAP-HN device types are defined in Device.  
**Returns:**  
type of this device

### getVersion

java.lang.String **getVersion()**  
Returns the version number associated with this Device's device type.  
**Returns:**  
a String representing the version of this Device's device type

### isLocal

boolean **isLocal()**  
Returns true when this is the local device.  
**Returns:**  
true if this is the local device

### addDeviceEventListener

void **addDeviceEventListener**(DeviceEventListener listener)  
Adds a DeviceEventListener instance to this Device. If the listener passed in is already registered with this Device, this method does nothing.  
**Parameters:**  
listener - a DeviceEventListener instance to be notified of DeviceEvents.

### removeDeviceEventListener

void **removeDeviceEventListener**(DeviceEventListener listener)  
Removes a DeviceEventListener instance from this Device. If the specified instance is not registered with this Device, this method does nothing.  
**Parameters:**  
listener - a DeviceEventListener instance to be removed from this Device.

### getInetAddress

java.net.InetAddress **getInetAddress()**  
Returns the IP address for this device.  
**Returns:**  
an InetAddress representing this device's IP address

### setFriendlyName

void **setFriendlyName**(java.lang.String value)  
Sets the value of the PROP\_FRIENDLY\_NAME property.  
**Parameters:**  
value - The value to set the property to.  
**Throws:**

`java.lang.IllegalArgumentException` - if the parameter violates the format specified by protocol mapping.

`java.lang.UnsupportedOperationException` - if the Device is not local; see the `isLocal` method.

`java.lang.SecurityException` - if the calling application has not been granted `HomeNetPermission("contentmanagement")`.

## org.ocap.hn Class DeviceEvent

```
java.lang.Object
├─ java.util.EventObject
│   └─ org.ocap.hn.DeviceEvent
```

### All Implemented Interfaces:

```
java.io.Serializable
```

```
public class DeviceEvent
extends java.util.EventObject
```

Represents a Device Event. There are two types of Device events: one that is generated by the NetManager when a Device is added or removed from the home network. Application may register as a listener to NetManager to receive such events. The other DeviceEvent is generated by the Device itself when its internal state changes. Application should register as a listener with a particular Device for such events. In both scenarios, the Device that was the source of the event is returned.

### See Also:

Serialized Form

## Field Summary

static int	<b>DEVICE_ADDED</b> A constant indicating new device is registered to home network.
static int	<b>DEVICE_REMOVED</b> A constant indicating a device is removed from home network.
static int	<b>DEVICE_UPDATED</b> A constant indicating a device is updated from home network.
static int	<b>STATE_CHANGE</b> A constant indicating a device's internal state has changed.

## Fields inherited from class java.util.EventObject

source

## Constructor Summary

**DeviceEvent**(int type, java.lang.Object source)  
Constructs a DeviceEvent by specifying type and source.

## Method Summary

java.lang.Object	<b>getSource</b> () Returns device event source, which is always a Device.
int	<b>getType</b> () Returns device event type, as defined in DeviceEvent.

**Methods inherited from class java.util.EventObject**

toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Field Detail****DEVICE\_ADDED**public static final int **DEVICE\_ADDED**

A constant indicating new device is registered to home network.

**See Also:**

Constant Field Values

**DEVICE\_REMOVED**public static final int **DEVICE\_REMOVED**

A constant indicating a device is removed from home network.

**See Also:**

Constant Field Values

**DEVICE\_UPDATED**public static final int **DEVICE\_UPDATED**

A constant indicating a device is updated from home network.

**See Also:**

Constant Field Values

**STATE\_CHANGE**public static final int **STATE\_CHANGE**

A constant indicating a device's internal state has changed.

**See Also:**

Constant Field Values

**Constructor Detail****DeviceEvent**public **DeviceEvent**(int type,  
                    java.lang.Object source)

Constructs a DeviceEvent by specifying type and source.

**Parameters:**

type - Device change type, allowed type are defined in DeviceEvent

source - Device where the change happens.

## Method Detail

### **getType**

```
public int getType()
```

Returns device event type, as defined in DeviceEvent.

**Returns:**

device event type

### **getSource**

```
public java.lang.Object getSource()
```

Returns device event source, which is always a Device.

**Overrides:**

getSource in class java.util.EventObject

**Returns:**

device event source

org.ocap.hn

## Interface DeviceEventListener

### All Superinterfaces:

java.util.EventListener

```
public interface DeviceEventListener
extends java.util.EventListener
```

DeviceEvent callback interface. When a Device is registered or removed from NetManager, or if the internal status of a Device changes, then system will notify all registered listeners.

## Method Summary

void	<b>notify</b> (DeviceEvent event)
	Callback function for Device events.

## Method Detail

### notify

```
void notify(DeviceEvent event)
    Callback function for Device events.
```

#### Parameters:

event - Device event

**org.ocap.hn**  
**Class HomeNetPermission**

```
java.lang.Object
├ java.security.Permission
│   └ java.security.BasicPermission
│       └ org.ocap.hn.HomeNetPermission
```

**All Implemented Interfaces:**  
 java.io.Serializable, java.security.Guard

```
public final class HomeNetPermission
extends java.security.BasicPermission
```

The HomeNetPermission class represents permission to execute privileged home networking operations only signed applications MAY be granted.

A HomeNetPermission consists of a permission name, representing a single privileged operation. The name given in the constructor may end in "\*" to represent all permissions beginning with the given string, such as "\*" to allow all HomeNetPermission operations.

The following table lists all HomeNetPermission permission names.

Permission Name	What the Permission Allows	Description
contentmanagement	Provides management of local or remote content	Applications with this permission can copy, move, delete content as well as allocate and delete logical volumes on a local network device.
contentlisting	Provides listing of content on remote devices	Applications with this permission can discover and query lists of content stored on or streamable from remote devices.
recording	Provides recording operations on remote devices	Applications with this permission can request that recordings be scheduled, prioritized, and deleted on remote devices.
recordinghandler	Provides recording request handler functionality on the local device	Applications with this permission can manage network recording requests for the local device.

Other permissions may be added as necessary.

**See Also:**  
 Serialized Form

<b>Constructor Summary</b>
<b>HomeNetPermission</b> ( java.lang.String name) Constructor for the HomeNetPermission

## Method Summary

### Methods inherited from class `java.security.BasicPermission`

`equals`, `getActions`, `hashCode`, `implies`, `newPermissionCollection`

### Methods inherited from class `java.security.Permission`

`checkGuard`, `getName`, `toString`

### Methods inherited from class `java.lang.Object`

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

### **HomeNetPermission**

```
public HomeNetPermission(java.lang.String name)
```

Constructor for the HomeNetPermission

**Parameters:**

name - The name of this permission (see table in class description).

## org.ocap.hn Class NetActionEvent

```
java.lang.Object
├─ java.util.EventObject
│   └─ org.ocap.hn.NetActionEvent
```

### All Implemented Interfaces:

```
java.io.Serializable
```

```
public class NetActionEvent
extends java.util.EventObject
```

This class represents an event generated in response to an action. NetActionEvent instances can only be created by the implementation.

### See Also:

Serialized Form

## Field Summary

static int	<b>ACTION_CANCELED</b> ACTION_CANCELED is returned by <code>getActionStatus()</code> when the action has been canceled.
static int	<b>ACTION_COMPLETED</b> Action status for a completed action
static int	<b>ACTION_FAILED</b> ACTION_FAILED is returned by <code>getActionStatus()</code> when the action has failed.
static int	<b>ACTION_IN_PROGRESS</b> ACTION_IN_PROGRESS is returned by <code>getActionStatus()</code> when the action is currently on going.
static int	<b>ACTION_STATUS_NOT_AVAILABLE</b> ACTION_STATUS_NOT_AVAILABLE is returned by <code>getActionStatus()</code> when the transaction has completed successfully or failed sometime before this method was called and the implementation is no longer maintaining a status for it.

## Fields inherited from class java.util.EventObject

source

## Constructor Summary

protected	<b>NetActionEvent</b> (java.lang.Object request, java.lang.Object response, int error, int status) Two argument constructor.
-----------	---

## Method Summary

NetActionRequest	<b>getActionRequest()</b> Returns the ActionRequest which identifies the action instance.
int	<b>getActionStatus()</b> Returns the status of the requested action.
int	<b>getError()</b> Gets the error value when getActionStatus returns NetActionEvent.ACTION_FAILED.
java.lang.Object	<b>getResponse()</b> Returns the response of the Action.

### Methods inherited from class java.util.EventObject

getSource, toString

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### ACTION\_COMPLETED

```
public static final int ACTION_COMPLETED
    Action status for a completed action
See Also:
    getActionStatus(), Constant Field Values
```

### ACTION\_CANCELED

```
public static final int ACTION_CANCELED
    ACTION_CANCELED is returned by getActionStatus() when the action has been canceled.
See Also:
    getActionStatus(), Constant Field Values
```

### ACTION\_FAILED

```
public static final int ACTION_FAILED
    ACTION_FAILED is returned by getActionStatus() when the action has failed.
See Also:
    getActionStatus(), Constant Field Values
```

### ACTION\_STATUS\_NOT\_AVAILABLE

```
public static final int ACTION_STATUS_NOT_AVAILABLE
    ACTION_STATUS_NOT_AVAILABLE is returned by getActionStatus() when the transaction has
    completed successfully or failed sometime before this method was called and the implementation is no
    longer maintaining a status for it.
See Also:
    Constant Field Values
```

## ACTION\_IN\_PROGRESS

```
public static final int ACTION_IN_PROGRESS
```

ACTION\_IN\_PROGRESS is returned by `getActionStatus()` when the action is currently on going.

**See Also:**

`getActionStatus()`, Constant Field Values

## Constructor Detail

### NetActionEvent

```
protected NetActionEvent( java.lang.Object request,
                           java.lang.Object response,
                           int error,
                           int status)
```

Two argument constructor.

**Parameters:**

`request` - - NetActionRequest that instigated the response.

`response` - - An object representing the response to the action and which is specific to the action.

`error` - - error code for this event if action failed

`status` - - status of the associated net action

## Method Detail

### getResponse

```
public java.lang.Object getResponse()
```

Returns the response of the Action. Object is dependent on the Action.

**Returns:**

The response to an asynchronous action.

### getRequest

```
public NetActionRequest getRequest()
```

Returns the ActionRequest which identifies the action instance.

**Returns:**

the ActionRequest

### getStatus

```
public int getStatus()
```

Returns the status of the requested action.

**Returns:**

the status of the action; for possible return values see ACTION\_\* constants in this class.

### getError

```
public int getError()
```

Gets the error value when `getStatus` returns `NetActionEvent.ACTION_FAILED`. If the action is not in error this method SHALL return -1. Error code values are dependent on the underlying network protocol error code values.

**Returns:**

The error value; -1 if no error.

**org.ocap.hn**  
**Interface NetActionHandler**

```
public interface NetActionHandler
```

This interface represents a handler passed to asynchronous methods

## Method Summary

void	<b>notify</b> (NetActionEvent event) Notifies the application of an action event.
------	--

## Method Detail

### **notify**

```
void notify(NetActionEvent event)
```

Notifies the application of an action event. This method is called by the implementation when a response to an action or a failure for the action is detected.

**org.ocap.hn****Interface NetActionRequest**

```
public interface NetActionRequest
```

All asynchronous actions in the Home networking API return an `NetActionRequest`. The `NetActionRequest` can be used a) to cancel any pending action or b) to identify which Action got completed.

**See Also:**

`NetActionHandler`, `NetActionEvent`

## Method Summary

boolean	<b>cancel()</b> Cancels the Action associated with this ActionRequest.
int	<b>getActionStatus()</b> Gets the current status of the requested action.
int	<b>getError()</b> Gets the error value when <code>getActionStatus</code> returns <code>NetActionEvent.ACTION_FAILED</code> .
float	<b>getProgress()</b> Gets the progress of the action in percent (0.0 - 1.0).

## Method Detail

**cancel**

```
boolean cancel()
```

Cancels the Action associated with this ActionRequest. Returns false if the action can't be canceled.

**Returns:**

false if action can't be canceled, otherwise returns true.

**getProgress**

```
float getProgress()
```

Gets the progress of the action in percent (0.0 - 1.0). If the progress of an action can't be determined, -1.0 shall be returned.

**Returns:**

the progress of the action (0.0 - 1.0) or -1.0 if the progress can't be determined.

**getActionStatus**

```
int getActionStatus()
```

Gets the current status of the requested action.

**Returns:**

the current action status; see `ACTION_*` constants in `NetActionEvent` for possible return values.

**getError**

```
int getError()
```

Gets the error value when `getActionStatus` returns `NetActionEvent.ACTION_FAILED`. The error code returned will be equivalent to the error code returned by `NetActionEvent.getError()` for the `NetActionEvent` associated with the completion of this action request. If the action is not in error or has not completed, this method SHALL return -1.

**Returns:**

The error value; -1 if no error,

## org.ocap.hn Interface NetList

```
public interface NetList
```

A list comprising of homenetwork elements such as Device or NetModule. The application may retrieve such a list from NetManager, getNetModules \* or getDevices. The application may refine the list by applying a PropertyFilter.

### Method Summary

boolean	<b>contains</b> (java.lang.Object element) Indicates whether an element is included in this NetList.
NetList	<b>filterElement</b> (PropertyFilter filter) Applies a new PropertyFilter to this element list and returns a new list.
java.lang.Object	<b>getElement</b> (int index) Returns the element indexed by a number.
java.util.Enumeration	<b>getElements</b> () Returns all elements in this NetList in Enumeration.
int	<b>indexOf</b> (java.lang.Object element) Returns the index of an element in this element list.
int	<b>size</b> () Returns the size of this list.

### Method Detail

#### contains

```
boolean contains(java.lang.Object element)
```

Indicates whether an element is included in this NetList.

**Parameters:**  
element - the element to check whether it is in the list

**Returns:**  
true if the element is in the list; otherwise false.

#### getElement

```
java.lang.Object getElement(int index)
```

Returns the element indexed by a number.

**Parameters:**  
index - specified index of the element

**Returns:**  
element indexed by the number

#### getElements

```
java.util.Enumeration getElements()
```

Returns all elements in this NetList in Enumeration. In Homenetwork, NetList can be used to retrieve a list of Devices or a list of NetModules. In either case, a corresponding type of object is returned.

**Returns:**

An enumeration of Device or NetModule elements

### filterElement

NetList **filterElement**(PropertyFilter filter)

Applies a new PropertyFilter to this element list and returns a new list.

**Parameters:**

filter - new filter

**Returns:**

new element list generated by new filter

**See Also:**

PropertyFilter

### indexOf

int **indexOf**(java.lang.Object element)

Returns the index of an element in this element list.

**Parameters:**

element - to be checked

**Returns:**

index of an element in this list. If there is no such element in this list, returns -1.

### size

int **size**()

Returns the size of this list.

**Returns:**

size of the element list

## org.ocap.hn Class NetManager

```
java.lang.Object
└─ org.ocap.hn.NetManager
```

```
public abstract class NetManager
extends java.lang.Object
```

The NetManager is a singleton class that registers all the Devices and NetModules within a home network. It maintains an implementation dependent database of devices and NetModules.

The NetManager may be used to retrieve list of NetModule and Device in the network. The application can filter the list by specifying a name or by applying filtering rules. For example, "modelName = h6315, location = LivingRoom". Application can monitor availability of NetModules by registering as a listener to NetManager instance.

### Constructor Summary

**NetManager**( )

### Method Summary

abstract void	<b>addDeviceEventListener</b> (DeviceEventListener listener) Adds a Device event listener to NetManager.
abstract void	<b>addNetModuleEventListener</b> (NetModuleEventListener listener) Adds a NetModule event listener to NetManager.
abstract Device	<b>getDevice</b> (java.lang.String name) Returns device by name.
abstract NetList	<b>getDeviceList</b> (PropertyFilter filter) Returns devices that match all properties set by a given filter.
static NetManager	<b>getInstance</b> ( ) Returns the singleton NetManager.
abstract NetModule	<b>getNetModule</b> (java.lang.String deviceName, java.lang.String moduleID) Returns NetModule by device and module ID.
abstract NetList	<b>getNetModuleList</b> (PropertyFilter filter) Returns NetModules that match all properties set by a given filter.
abstract void	<b>removeDeviceEventListener</b> (DeviceEventListener listener) Removes a Device event listener from NetManager.
abstract void	<b>removeNetModuleEventListener</b> (NetModuleEventListener listener) Removes a NetModule event listener from NetManager.

## Method Summary

abstract void	<b>updateDeviceList()</b> Requests that the NetManager proactively refresh its local database of connected devices.
---------------	--

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### NetManager

```
public NetManager()
```

## Method Detail

### getInstance

```
public static NetManager getInstance()
```

Returns the singleton NetManager. This is the entry point for home network. If the calling application is unsigned, this method SHALL return null.

**Returns:**

Singleton instance of NetManager or null if the calling application is unsigned.

### getNetModuleList

```
public abstract NetList getNetModuleList(PropertyFilter filter)
```

Returns NetModules that match all properties set by a given filter. Passing a null filter will return a NetList with all known NetModules.

**Parameters:**

*filter* - Filter to select out NetModules from all available NetModules

**Returns:**

List of NetModules satisfying filter

### getNetModule

```
public abstract NetModule getNetModule(java.lang.String deviceName,
                                       java.lang.String moduleID)
```

Returns NetModule by device and module ID.

**Parameters:**

*deviceName* - name of the device hosting the module to retrieve

*moduleID* - Device unique module ID

**Returns:**

NetModule with the specified identifier

### getDeviceList

```
public abstract NetList getDeviceList(PropertyFilter filter)
```

Returns devices that match all properties set by a given filter. All known devices and sub-devices are passed through the given filter. Passing a null filter will return a NetList with all known devices and sub-devices.

**Parameters:**

filter - Filter to select out devices from all connected devices

**Returns:**

List of devices satisfying filter

### getDevice

```
public abstract Device getDevice(java.lang.String name)
```

Returns device by name. This name is unique device name. For example, "BallRoom:DVD\_PLAYER1".

**Parameters:**

name - Device name

**Returns:**

Device matching the specified name

### addNetModuleEventListener

```
public abstract void
```

```
addNetModuleEventListener(NetModuleEventListener listener)
```

Adds a NetModule event listener to NetManager. Listener will receive a NetModuleEvent when a new NetModule is registered or an old NetModule is removed from home network. If listener is already registered, no action is performed.

**Parameters:**

listener - Listener which listens to NetModule change events on home network

**See Also:**

removeNetModuleEventListener(org.ocap.hn.NetModuleEventListener)

### removeNetModuleEventListener

```
public abstract void
```

```
removeNetModuleEventListener(NetModuleEventListener listener)
```

Removes a NetModule event listener from NetManager. If the listener is not registered yet, no action is performed.

**Parameters:**

listener - Listener which listens to NetModule change events on home network

**See Also:**

addNetModuleEventListener(org.ocap.hn.NetModuleEventListener)

### addDeviceEventListener

```
public abstract void addDeviceEventListener(DeviceEventListener listener)
```

Adds a Device event listener to NetManager. Listener will receive a DeviceEvent when a new Device is registered, an existing Device is removed from home network, or a Device's internal state has changed. If the listener passed in is already registered, no action is performed. When a device listener is registered, the implementation SHALL NOT generate DEVICE\_ADDED events for devices previously discovered by the implementation.

**Parameters:**

listener - Listener which listens to Device change events on the home network

**See Also:**

removeDeviceEventListener(org.ocap.hn.DeviceEventListener)

**removeDeviceEventListener**

public abstract void **removeDeviceEventListener**(DeviceEventListener listener)  
Removes a Device event listener from NetManager. If the listener is not registered yet, no action is performed.

**Parameters:**

listener - Listener which listens to Device change events on home network

**See Also:**

`addDeviceEventListener(org.ocap.hn.DeviceEventListener)`

**updateDeviceList**

public abstract void **updateDeviceList**()

Requests that the NetManager proactively refresh its local database of connected devices. This operation will be performed asynchronously. Any listeners registered with the NetManager changes to connected Devices or NetModules will be notified of any changes discovered during this process.

**org.ocap.hn**  
**Interface NetModule**

**All Known Subinterfaces:**

ContentServerNetModule, NetRecordingRequestManager, RecordingNetModule

public interface **NetModule**

NetModule is an abstraction of functionality that is provided by a Device. It is a group of related actions. A NetModule is always associated with a homenetwork Device. Application may monitor a NetModule's status by subscribing as a listener to this NetModule.

**Field Summary**

static java.lang.String	<b>CONTENT_LIST</b> A constant indicating content listing NetModule.
static java.lang.String	<b>CONTENT_MANAGER</b> A constant indicating content manager NetModule.
static java.lang.String	<b>CONTENT_RECORDER</b> A constant indicating content recording NetModule.
static java.lang.String	<b>CONTENT_RENDERER</b> A constant indicating content renderer NetModule.
static java.lang.String	<b>CONTENT_SERVER</b> A constant indicating content server NetModule.
static java.lang.String	<b>PROP_CONTROL_URL</b> A constant providing URL for NetModule control.
static java.lang.String	<b>PROP_DESCRIPTION_URL</b> A constant providing URL for NetModule description.
static java.lang.String	<b>PROP_EventSub_URL</b> A constant providing URL for NetModule eventing.
static java.lang.String	<b>PROP_NETMODULE_ID</b> A constant indicating NetModuleID.
static java.lang.String	<b>PROP_NETMODULE_TYPE</b> A constant providing this NetModule's type.

**Method Summary**

void	<b>addNetModuleEventListener</b> (NetModuleEventListener listener) Adds a NetModuleEventListener instance to this NetModule.
Device	<b>getDevice</b> () Returns the device that provides this NetModule.
java.util.Enumeration	<b>getKeys</b> () Returns the property keys supported by this NetModule.

## Method Summary

java.lang.String	<b>getNetModuleId()</b> Returns the id of this NetModule, which is unique within the device.
java.lang.String	<b>getNetModuleType()</b> Returns the type of this NetModule.
java.lang.String	<b>getProperty</b> (java.lang.String key) Returns the property value for specified key.
boolean	<b>isLocal()</b> Returns true if this NetModule is hosted on the local device.
void	<b>removeNetModuleEventListener</b> (NetModuleEventListener listener) Removes a NetModuleEventListener instance from this NetModule.

## Field Detail

### CONTENT\_LIST

static final java.lang.String **CONTENT\_LIST**  
A constant indicating content listing NetModule.  
**See Also:**  
Constant Field Values

### CONTENT\_MANAGER

static final java.lang.String **CONTENT\_MANAGER**  
A constant indicating content manager NetModule.  
**See Also:**  
Constant Field Values

### CONTENT\_RENDERER

static final java.lang.String **CONTENT\_RENDERER**  
A constant indicating content renderer NetModule.  
**See Also:**  
Constant Field Values

### CONTENT\_SERVER

static final java.lang.String **CONTENT\_SERVER**  
A constant indicating content server NetModule.  
**See Also:**  
Constant Field Values

### CONTENT\_RECORDER

static final java.lang.String **CONTENT\_RECORDER**  
A constant indicating content recording NetModule.  
**See Also:**  
Constant Field Values

**PROP\_NETMODULE\_ID**

static final java.lang.String **PROP\_NETMODULE\_ID**  
A constant indicating NetModuleID.  
**See Also:**  
Constant Field Values

**PROP\_DESCRIPTION\_URL**

static final java.lang.String **PROP\_DESCRIPTION\_URL**  
A constant providing URL for NetModule description.  
**See Also:**  
Constant Field Values

**PROP\_CONTROL\_URL**

static final java.lang.String **PROP\_CONTROL\_URL**  
A constant providing URL for NetModule control.  
**See Also:**  
Constant Field Values

**PROP\_EventSub\_URL**

static final java.lang.String **PROP\_EventSub\_URL**  
A constant providing URL for NetModule eventing.  
**See Also:**  
Constant Field Values

**PROP\_NETMODULE\_TYPE**

static final java.lang.String **PROP\_NETMODULE\_TYPE**  
A constant providing this NetModule's type.  
**See Also:**  
Constant Field Values

## Method Detail

**getDevice**

Device **getDevice()**  
Returns the device that provides this NetModule.  
**Returns:**  
device that offers this NetModule

**getKeys**

java.util.Enumeration **getKeys()**  
Returns the property keys supported by this NetModule.  
**Returns:**  
An enumeration of String object representing property keys for this NetModule

**getProperty**

java.lang.String **getProperty**(java.lang.String key)  
Returns the property value for specified key.

**Parameters:**

key - specified property key

**Returns:**

property value for specified key

**getNetModuleType**

```
java.lang.String getNetModuleType()
```

Returns the type of this NetModule. The allowed types are defined as constant field in NetModule, for example, CONTENT\_MANAGER, CONTENT\_LIST.

**Returns:**

type of this NetModule

**getNetModuleId**

```
java.lang.String getNetModuleId()
```

Returns the id of this NetModule, which is unique within the device. An example could be, ContentListing1.

**Returns:**

id of this NetModule

**addNetModuleEventListener**

```
void addNetModuleEventListener(NetModuleEventListener listener)
```

Adds a NetModuleEventListener instance to this NetModule. If the listener passed in is already registered with this NetModule, this method does nothing.

**Parameters:**

listener - a NetModuleEventListener instance to be notified of NetModuleEvents.

**removeNetModuleEventListener**

```
void removeNetModuleEventListener(NetModuleEventListener listener)
```

Removes a NetModuleEventListener instance from this NetModule. If the specified instance is not registered with this NetModule, this method does nothing.

**Parameters:**

listener - a NetModuleEventListener instance to be removed from this NetModule.

**isLocal**

```
boolean isLocal()
```

Returns true if this NetModule is hosted on the local device.

**Returns:**

true if this NetModule is hosted on the local device, false otherwise.

**org.ocap.hn****Class NetModuleEvent**

```

java.lang.Object
├─ java.util.EventObject
│   └─ org.ocap.hn.NetModuleEvent

```

**All Implemented Interfaces:**

```

java.io.Serializable

```

```

public class NetModuleEvent
extends java.util.EventObject

```

Entity for NetModule Event. There are two types of NetModule events: one that is generated by the NetManager when a NetModule is added or removed from the home network. Application may register as a listener to NetManager to receive such events. The other NetModuleEvent is generated by the NetModule itself when its internal state changes. Application should register as a listener with a particular NetModule for such events. In both scenarios, the NetModule that was the source of the event is returned.

**See Also:**

Serialized Form

**Field Summary**

static int	<b>MODULE_ADDED</b> A constant indicating new module is registered to home network.
static int	<b>MODULE_BUSY</b> A constant indicating a module is busy and cannot respond to request now.
static int	<b>MODULE_REMOVED</b> A constant indicating a module is removed from home network.
static int	<b>MODULE_UPDATED</b> A constant indicating a module is updated from home network.
static int	<b>STATE_CHANGE</b> A constant indicating a module's internal status changed.

**Fields inherited from class java.util.EventObject**

source

**Constructor Summary**

```

NetModuleEvent(int type, java.lang.Object source)
Constructs a NetModuleEvent by specifying type and source.

```

**Method Summary**

java.lang.Object	<b>getSource()</b> Returns module event source, which is always a NetModule.
------------------	---

## Method Summary

int	<b>getType()</b> Returns module event type, as defined in NetModuleEvent.
-----	--

### Methods inherited from class java.util.EventObject

toString

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### MODULE\_ADDED

```
public static final int MODULE_ADDED
```

A constant indicating new module is registered to home network.  
**See Also:**  
 Constant Field Values

### MODULE\_REMOVED

```
public static final int MODULE_REMOVED
```

A constant indicating a module is removed from home network.  
**See Also:**  
 Constant Field Values

### MODULE\_UPDATED

```
public static final int MODULE_UPDATED
```

A constant indicating a module is updated from home network.  
**See Also:**  
 Constant Field Values

### MODULE\_BUSY

```
public static final int MODULE_BUSY
```

A constant indicating a module is busy and cannot respond to request now.  
**See Also:**  
 Constant Field Values

### STATE\_CHANGE

```
public static final int STATE_CHANGE
```

A constant indicating a module's internal status changed.  
**See Also:**  
 Constant Field Values

## Constructor Detail

### NetModuleEvent

```
public NetModuleEvent(int type,  
                      java.lang.Object source)
```

Constructs a NetModuleEvent by specifying type and source.

**Parameters:**

type - NetModule change type, allowed type are defined in NetModuleEvent

source - NetModule where the change happens.

## Method Detail

### getType

```
public int getType()
```

Returns module event type, as defined in NetModuleEvent.

**Returns:**

module event type

### getSource

```
public java.lang.Object getSource()
```

Returns module event source, which is always a NetModule.

**Overrides:**

getSource in class java.util.EventObject

**Returns:**

module event source

org.ocap.hn

## Interface NetModuleEventListener

### All Superinterfaces:

java.util.EventListener

```
public interface NetModuleEventListener
extends java.util.EventListener
```

NetModuleEvent callback interface. When a NetModule is registered or removed from NetManager, or if the internal status of a NetModule changes, then system will notify all registered listeners.

## Method Summary

void	<b>notify</b> (NetModuleEvent event)
	Callback function for NetModule event.

## Method Detail

### notify

```
void notify(NetModuleEvent event)
```

Callback function for NetModule event. Callee will be notified when NetModule event happens

#### Parameters:

event - NetModule event

**org.ocap.hn****Class NetworkInterface**

```
java.lang.Object
└─ org.ocap.hn.NetworkInterface
```

```
public class NetworkInterface
extends java.lang.Object
```

This class represents a home network interface including MoCA, wired ethernet, and wireless ethernet. Reverse channel interfaces are not represented by objects of this class. For each wired ethernet, wireless ethernet, MoCA interface, or interface that is not a reverse channel interface the HNIMP SHALL create an instance of this class.

**Field Summary**

static int	<b>MOCA</b> Network interface type for hard-wired and MoCA based.
static int	<b>UNKNOWN</b> Unknown network type.
static int	<b>WIRED_ETHERNET</b> Network interface type for hard-wired and ethernet based.
static int	<b>WIRELESS_ETHERNET</b> Network interface type for wireless and ethernet based.

**Constructor Summary**

protected	<b>NetworkInterface()</b> Protected constructor.
-----------	---

**Method Summary**

java.lang.String	<b>getDisplayName()</b> Gets a humanly readable name for this interface, e.g.
java.net.InetAddress	<b>getInetAddress()</b> Gets the InetAddress of this interface.
java.lang.String	<b>getMacAddress()</b> Gets the MAC address of this interface.
static NetworkInterface[]	<b>getNetworkInterfaces()</b> Gets an array of NetworkInterface instances that represent all of the network interfaces supported by the device.
int	<b>getType()</b> Gets the type of this network interface.

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Field Detail****UNKNOWN**

public static final int **UNKNOWN**

Unknown network type.

**See Also:**

Constant Field Values

**MOCA**

public static final int **MOCA**

Network interface type for hard-wired and MoCA based.

**See Also:**

Constant Field Values

**WIRED\_ETHERNET**

public static final int **WIRED\_ETHERNET**

Network interface type for hard-wired and ethernet based.

**See Also:**

Constant Field Values

**WIRELESS\_ETHERNET**

public static final int **WIRELESS\_ETHERNET**

Network interface type for wireless and ethernet based.

**See Also:**

Constant Field Values

**Constructor Detail****NetworkInterface**

protected **NetworkInterface**()

Protected constructor.

**Method Detail****getNetworkInterfaces**

public static NetworkInterface[] **getNetworkInterfaces**()

Gets an array of NetworkInterface instances that represent all of the network interfaces supported by the device.

**Returns:**

An array of NetworkInterface instances.

**getType**

```
public int getType()
```

Gets the type of this network interface. Possibilities include UNKNOWN, MOCA, WIRED\_ETHERNET, WIRELESS\_ETHERNET.

**Returns:**

The type of this interface.

**getDisplayName**

```
public java.lang.String getDisplayName()
```

Gets a humanly readable name for this interface, e.g. "ie0".

**Returns:**

The display name of this interface.

**getInetAddress**

```
public java.net.InetAddress getInetAddress()
```

Gets the InetAddress of this interface.

**Returns:**

The InetAddress of this interface.

**getMacAddress**

```
public java.lang.String getMacAddress()
```

Gets the MAC address of this interface.

**Returns:**

The MAC address of this interface.

**org.ocap.hn****Class NotAuthorizedException**

```

java.lang.Object
├─ java.lang.Throwable
│   └─ java.lang.Exception
│       └─ org.ocap.hn.NotAuthorizedException

```

**All Implemented Interfaces:**

```

java.io.Serializable

```

```

public class NotAuthorizedException
extends java.lang.Exception

```

Exception indicating that the application has no permission to perform certain action.

**See Also:**

Serialized Form

**Constructor Summary**

**NotAuthorizedException**()

Constructs a NotAuthorizedException object.

**NotAuthorizedException**(java.lang.String reason)

Constructs a NotAuthorizedException object with a reason.

**Method Summary****Methods inherited from class java.lang.Throwable**

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Constructor Detail****NotAuthorizedException**

```

public NotAuthorizedException()

```

Constructs a NotAuthorizedException object.

**NotAuthorizedException**

```

public NotAuthorizedException(java.lang.String reason)

```

Constructs a NotAuthorizedException object with a reason.

**Parameters:**

reason - reason for this exception

## org.ocap.hn Class PropertyFilter

```
java.lang.Object
└─ org.ocap.hn.PropertyFilter
```

```
public class PropertyFilter
extends java.lang.Object
```

The filter for (key,value) pair filtering mechanism. If a device or a NetModule has same value on all of the specified keys, it is regarded as a match.

### Constructor Summary

<b>PropertyFilter</b> ( java.util.Properties prop)	Constructs a PropertyFilter object.
--	-------------------------------------

### Method Summary

boolean	<b>accept</b> ( java.lang.Object element) Checks whether an element is accepted by this filter, the element must be either NetModule or Device.
void	<b>addProperty</b> ( java.lang.String key, java.lang.String value) Adds a (key,value) pair to the filter.
boolean	<b>contains</b> ( java.lang.String key) Checks whether a key is in the list.
void	<b>removeKey</b> ( java.lang.String key) Remove a key from the filter, if the key is not in the property list, no action is taken.
void	<b>removeKeys</b> ( java.lang.String[] keys) Remove keys from the filter, if a key is not in the property list, it is disregarded; while others are processed as normal.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### PropertyFilter

```
public PropertyFilter( java.util.Properties prop)
    Constructs a PropertyFilter object.
Parameters:
    prop - Initial properties for this Property filter
```

## Method Detail

### addProperty

```
public void addProperty(java.lang.String key,  
                        java.lang.String value)
```

Adds a (key,value) pair to the filter. If the key is already in the list, no action is taken.

**Parameters:**

key - New key which will be used for filtering.

value - Value for the new key.

### contains

```
public boolean contains(java.lang.String key)
```

Checks whether a key is in the list.

**Parameters:**

key - Key to be checked against.

**Returns:**

True if key is in the list; otherwise returns false.

### accept

```
public boolean accept(java.lang.Object element)
```

Checks whether an element is accepted by this filter, the element must be either NetModule or Device.

If a NetModule/Device's properties share the same value as all properties from this filter, it is accepted and true is returned; otherwise, false is returned.

**Parameters:**

element - Element to be checked against.

**Returns:**

True if the element is accepted by the PropertyFilter, otherwise returns false.

### removeKey

```
public void removeKey(java.lang.String key)
```

Remove a key from the filter, if the key is not in the property list, no action is taken.

**Parameters:**

key - Key to be removed from list.

### removeKeys

```
public void removeKeys(java.lang.String[] keys)
```

Remove keys from the filter, if a key is not in the property list, it is disregarded; while others are processed as normal.

**Parameters:**

keys - Keys to be removed from the list.

## Annex B Content API

### Package org.ocap.hn.content

#### Interface Summary

<b>AudioResource</b>	Interface implemented by subclasses of ContentResource to identify that a content contains audio.
<b>ContentContainer</b>	This class represents a container that contains one or more content entries.
<b>ContentEntry</b>	This interface represents a basic content entry.
<b>ContentItem</b>	This class represents a piece of content.
<b>ContentResource</b>	Abstract class representing a media stream/file.
<b>IOStatus</b>	This interface represents the ability to detect whether any asset represented by an object or its children is in use on the home network and hence the object should not be deleted.
<b>StreamableContentResource</b>	Abstract class representing content that can be streamed, e.g., MPEG file.
<b>VideoResource</b>	ContentResource to identify that a content item contains video/still image material.

#### Class Summary

<b>MetadataIdentifiers</b>	This abstract class represents access to standardized metadata identifiers.
<b>MetadataNode</b>	Base class for all Metadata.

#### Exception Summary

<b>DatabaseException</b>	Exception that is thrown when a database error occurs
--------------------------	---

## org.ocap.hn.content Interface AudioResource

**All Superinterfaces:**  
ContentResource

```
public interface AudioResource
extends ContentResource
```

Interface implemented by subclasses of ContentResource to identify that a content contains audio.

### Field Summary

Fields inherited from interface org.ocap.hn.content.ContentResource

UNKNOWN\_MIME\_TYPE

### Method Summary

int	<b>getBitsPerSample()</b> Returns the number of bits per sample or -1 if not known.
java.lang.String[]	<b>getLanguages()</b> Returns an array of languages associated with this audio content or a zero length array if not known.
int	<b>getNumberOfChannels()</b> Returns the number of audio channels, for example, 1 for mono, 2 for stereo, 6 for DTS 5.1 and 7 for DTS 6.1
int	<b>getSampleFrequency()</b> Returns the sample frequency in Hz of this audio content or -1 if not known.

Methods inherited from interface org.ocap.hn.content.ContentResource

delete, getContentFormat, getContentItem, getContentSize, getCreationDate, getExtendedFileAccessPermissions, getLocator, getNetwork, getProtocol, getResourceProperty, isRenderable

### Method Detail

#### getSampleFrequency

```
int getSampleFrequency()
    Returns the sample frequency in Hz of this audio content or -1 if not known.
Returns:
    the sample frequency of the content of -1 if not known.
```

#### getNumberOfChannels

```
int getNumberOfChannels()
```

Returns the number of audio channels, for example, 1 for mono, 2 for stereo, 6 for DTS 5.1 and 7 for DTS 6.1

**Returns:**

the sample frequency of the content of -1 if not known.

**getBitsPerSample**

int **getBitsPerSample()**

Returns the number of bits per sample or -1 if not known.

**Returns:**

the number of bits per sample or -1 if not known.

**getLanguages**

java.lang.String[] **getLanguages()**

Returns an array of languages associated with this audio content or a zero length array if not known.

**Returns:**

the languages associated with this audio.

**org.ocap.hn.content**  
**Interface ContentContainer**

**All Superinterfaces:**  
 ContentEntry

```
public interface ContentContainer
extends ContentEntry
```

This class represents a container that contains one or more content entries. Can contain children containers.

<b>Field Summary</b>	
static java.lang.String	<b>ALBUM_CONTAINER</b> Represents the base album container.
static java.lang.String	<b>ALBUM_CONTAINER_MUSIC</b> Represents a music album container.
static java.lang.String	<b>ALBUM_CONTAINER_PHOTO</b> Represents a photo album container.
static java.lang.String	<b>CONTAINER</b> Represents the base container class.
static java.lang.String	<b>GENRE_CONTAINER</b> Represents an unordered collection of 'objects' that "belong" to the genre.
static java.lang.String	<b>GENRE_CONTAINER_MOVIE</b> Represents a movie genre container.
static java.lang.String	<b>GENRE_CONTAINER_MUSIC</b> Represents a music genre container.
static java.lang.String	<b>PERSON_CONTAINER</b> Represents an unordered collection of 'objects' that "belong" to the people.
static java.lang.String	<b>PERSON_CONTAINER_MUSIC_ARTIST</b> Represents a music artist person container.
static java.lang.String	<b>PLAYLIST_CONTAINER</b> Represents a collection of objects.
static java.lang.String	<b>STORAGE_FOLDER_CONTAINER</b> Represents all, or a partition of some physical storage unit of a single type.
static java.lang.String	<b>STORAGE_SYSTEM_CONTAINER</b> Represents a potentially heterogeneous collection of storage media.
static java.lang.String	<b>STORAGE_VOLUME_CONTAINER</b> Represents all, or a partition of, some physical storage unit of a single type.

<b>Method Summary</b>	
boolean	<b>addContentEntries</b> (ContentEntry[] entries) Adds ContentEntry objects to this ContentContainer.
boolean	<b>addContentEntry</b> (ContentEntry entry) Adds a ContentEntry to this ContentContainer.
boolean	<b>contains</b> (ContentEntry entry) Checks whether the given ContentEntry is in this ContentContainer in local cache only.
boolean	<b>createContentContainer</b> (java.lang.String name, ExtendedFileAccessPermissions permissions) Creates a new ContentContainer as a child of this ContentContainer.
boolean	<b>createContentItem</b> (java.io.File content, java.lang.String name, ExtendedFileAccessPermissions permissions) Creates a new ContentItem representing a local file as a child of this ContentContainer.
boolean	<b>delete</b> () Deletes this ContentContainer if and only if it is empty.
boolean	<b>deleteContents</b> () Deletes all the ContentEntry objects in this container except for ContentContainer entries.
boolean	<b>deleteRecursive</b> (boolean recursive) Deletes this ContentContainer and all of its ContentEntry objects.
int	<b>getComponentCount</b> () Gets the number of ContentEntry objects in this ContentContainer.
java.lang.String	<b>getContainerClass</b> () Returns the container class of this container.
long	<b>getContentSize</b> () Gets the size of the ContentContainer and all its content including all its contained ContentContainer objects.
java.util.Date	<b>getCreationDate</b> () Returns the creation date of this ContentContainer.
java.util.Enumeration	<b>getEntries</b> () Gets an Enumeration over all entries in this ContentContainer, from local cache only; does not cause network activity.
ContentList	<b>getEntries</b> (ContentDatabaseFilter filter, boolean traverse) Returns a ContentList which contains the filtered ContentItems of this ContentContainer.
ContentEntry	<b>getEntry</b> (int n) Returns the n <sup>th</sup> ContentEntry in this container, from local cache only; does not cause network activity.
ContentEntry	<b>getEntry</b> (java.lang.String ID) Returns the ContentEntry associated with the given ID in this container, or NULL if no entry is found.

## Method Summary

ExtendedFileAccessPermissions	<b>getExtendedFileAccessPermissions()</b> Gets the ExtendedFileAccessPermissions of this ContentContainer.
int	<b>getIndex(ContentEntry n)</b> Gets the index of the specified ContentEntry, from local cache only; does not cause network activity.
java.lang.String	<b>getName()</b> Gets the name of this ContentContainer.
boolean	<b>isEmpty()</b> Returns an empty indication.
boolean	<b>removeContentEntries(ContentEntry[] entries)</b> Removes ContentEntry objects from this ContentContainer.
boolean	<b>removeContentEntry(ContentEntry entry)</b> Removes a ContentEntry from this ContentContainer.
ContentEntry[]	<b>toArray()</b> Returns an array of all ContentEntry in this ContentContainers including other ContentContainers.

### Methods inherited from interface org.ocap.hn.content.ContentEntry

deleteEntry, getEntryParent, getID, getParentID, getRootMetadataNode, getServer, isLocal

## Field Detail

### CONTAINER

static final java.lang.String **CONTAINER**

Represents the base container class.

**See Also:**

Constant Field Values

### ALBUM\_CONTAINER

static final java.lang.String **ALBUM\_CONTAINER**

Represents the base album container.

**See Also:**

Constant Field Values

### ALBUM\_CONTAINER\_PHOTO

static final java.lang.String **ALBUM\_CONTAINER\_PHOTO**

Represents a photo album container. In addition to being an ALBUM\_CONTAINER container may be a photo album.

**See Also:**

Constant Field Values

**ALBUM\_CONTAINER\_MUSIC**

```
static final java.lang.String ALBUM_CONTAINER_MUSIC
```

Represents a music album container. In addition to being an ALBUM\_CONTAINER container may be a music album.

**See Also:**

Constant Field Values

**GENRE\_CONTAINER**

```
static final java.lang.String GENRE_CONTAINER
```

Represents an unordered collection of 'objects' that "belong" to the genre.

**See Also:**

Constant Field Values

**GENRE\_CONTAINER\_MUSIC**

```
static final java.lang.String GENRE_CONTAINER_MUSIC
```

Represents a music genre container. In addition to being a GENRE\_CONTAINER a container may be a music genre container

**See Also:**

Constant Field Values

**GENRE\_CONTAINER\_MOVIE**

```
static final java.lang.String GENRE_CONTAINER_MOVIE
```

Represents a movie genre container. In addition to being a GENRE\_CONTAINER a container may be a movie genre container

**See Also:**

Constant Field Values

**PLAYLIST\_CONTAINER**

```
static final java.lang.String PLAYLIST_CONTAINER
```

Represents a collection of objects.

**See Also:**

Constant Field Values

**PERSON\_CONTAINER**

```
static final java.lang.String PERSON_CONTAINER
```

Represents an unordered collection of 'objects' that "belong" to the people.

**See Also:**

Constant Field Values

**PERSON\_CONTAINER\_MUSIC\_ARTIST**

```
static final java.lang.String PERSON_CONTAINER_MUSIC_ARTIST
```

Represents a music artist person container. In addition to being a PERSON\_CONTAINER a container may be a music artist.

**See Also:**

Constant Field Values

### STORAGE\_SYSTEM\_CONTAINER

static final java.lang.String **STORAGE\_SYSTEM\_CONTAINER**  
Represents a potentially heterogeneous collection of storage media.  
**See Also:**  
Constant Field Values

### STORAGE\_VOLUME\_CONTAINER

static final java.lang.String **STORAGE\_VOLUME\_CONTAINER**  
Represents all, or a partition of, some physical storage unit of a single type.  
**See Also:**  
Constant Field Values

### STORAGE\_FOLDER\_CONTAINER

static final java.lang.String **STORAGE\_FOLDER\_CONTAINER**  
Represents all, or a partition of some physical storage unit of a single type.  
**See Also:**  
Constant Field Values

## Method Detail

### getContainerClass

java.lang.String **getContainerClass()**  
Returns the container class of this container.  
**Returns:**  
The content class of this item.  
**See Also:**  
ALBUM\_CONTAINER, ALBUM\_CONTAINER\_MUSIC, ALBUM\_CONTAINER\_PHOTO,  
GENRE\_CONTAINER, GENRE\_CONTAINER\_MUSIC, GENRE\_CONTAINER\_MOVIE,  
PLAYLIST\_CONTAINER, PERSON\_CONTAINER, PERSON\_CONTAINER\_MUSIC\_ARTIST,  
STORAGE\_SYSTEM\_CONTAINER, STORAGE\_VOLUME\_CONTAINER,  
STORAGE\_FOLDER\_CONTAINER

### toArray

ContentEntry[] **toArray()**  
Returns an array of all ContentEntry in this ContentContainers including other ContentContainers. Returns ContentEntry objects stored in local cache only; does not cause network activity.  
**Returns:**  
array containing all entries of this ContentContainers

### contains

boolean **contains**(ContentEntry entry)  
Checks whether the given ContentEntry is in this ContentContainer in local cache only.  
**Parameters:**  
entry - To search for in this ContentEntry.  
**Returns:**  
True if the ContentEntry is contained in this container, otherwise returns false.

**getEntry**

ContentEntry **getEntry**(java.lang.String ID)

Returns the ContentEntry associated with the given ID in this container, or NULL if no entry is found. This method SHALL recursively search this container and any sub-containers. This method searches local cache only; does not cause network activity.

**Parameters:**

ID - String ID of the ContentEntry to return

**Returns:**

the associated ContentEntry.

**See Also:**

ContentEntry.getID()

**getEntry**

ContentEntry **getEntry**(int n)

Returns the n<sup>th</sup> ContentEntry in this container, from local cache only; does not cause network activity.

**Parameters:**

n - Index of the entry to get.

**Returns:**

the n<sup>th</sup> ContentEntry.

**Throws:**

java.lang.ArrayIndexOutOfBoundsException - if the n<sup>th</sup> value does not exist.

**getEntries**

java.util.Enumeration **getEntries**()

Gets an Enumeration over all entries in this ContentContainer, from local cache only; does not cause network activity.

**Returns:**

Enumeration over all entries in this ContentContainers, or null if there are no entries.

**getIndex**

int **getIndex**(ContentEntry n)

Gets the index of the specified ContentEntry, from local cache only; does not cause network activity.

**Parameters:**

n - The index of the ContentEntry to search for.

**Returns:**

The index of the ContentEntry or -1 if it doesn't exist in this container.

**createContentItem**

boolean **createContentItem**(java.io.File content,  
java.lang.String name,  
ExtendedFileAccessPermissions permissions)

Creates a new ContentItem representing a local file as a child of this ContentContainer. If this ContentContainer #isLocal method returns false this method will return false. The resulting ContentItem will contain a single ContentResource containing the content parameter passed to this method.

**Parameters:**

content - The file containing the content to be represented

name - The name of the new ContentItem.

permissions - Access permissions of the new ContentContainer.

**Returns:**

True if a new ContentContainers has been created, otherwise return false.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("contentmanagement"), or if the caller does not have write permission on this container.

**createContentContainer**

```
boolean createContentContainer(java.lang.String name,  
                               ExtendedFileAccessPermissions permissions)
```

Creates a new ContentContainer as a child of this ContentContainer. If this ContentContainer #isLocal method returns false this method will return false. Can be used to create a directory structure.

**Parameters:**

name - The name of the new ContentContainer.  
permissions - Access permissions of the new ContentContainer.

**Returns:**

True if a new ContentContainer has been created, otherwise returns false.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("contentmanagement"), or if the caller does not have write permission on this container.

**getEntries**

```
ContentList getEntries(ContentDatabaseFilter filter,  
                       boolean traverse)
```

Returns a ContentList which contains the filtered ContentItems of this ContentContainer. If the traverse parameter is true the ContentItems of all its children ContentContainers is included. The list returned is filtered by the filter parameter. If the filter is null all items are returned.

**Parameters:**

filter - A ContentDatabaseFilter to filter the ContentItems. If the filter is null all entries are returned  
traverse - If true entries in the sub-containers are returned, otherwise only entries in this ContentContainer are returned.

**Returns:**

a ContentList filtered by the ContentDatabaseFilter

**getName**

```
java.lang.String getName()  
    Gets the name of this ContentContainer.
```

**Returns:**

The name of this ContentContainer.

**See Also:**

ContentEntry.getID()

**delete**

```
boolean delete()  
    throws java.io.IOException
```

Deletes this ContentContainer if and only if it is empty. References to this ContentContainer are not valid after this call. This method deletes a local ContentContainer only. If the #isLocal method returns false an exception is thrown.

**Returns:**

True if this ContentContainer was deleted, otherwise returns false.

**Throws:**

java.lang.SecurityException - if the application is denied to perform the action

`java.io.IOException` - if this `ContentContainer` is not local.

### **deleteContents**

boolean **deleteContents**()

throws `java.io.IOException`

Deletes all the `ContentEntry` objects in this container except for `ContentContainer` entries. Calls the `ContentEntry.deleteEntry()` method on each entry in this container. References to `ContentEntry` objects in this container are not valid after this call when the deletions are successful. This method deletes local `ContentEntry` instances only. If the `#isLocal` method returns false, an exception is thrown.

**Returns:**

True if all of the `ContentEntry` objects in this container were deleted, otherwise returns false.

**Throws:**

`java.lang.SecurityException` - if the caller does not have

`HomeNetPermission("contentmanagement")`, or if the caller does not have write permission on this container or and entries contained in this container.

`java.io.IOException` - if this `ContentContainer` is not local.

### **deleteRecursive**

boolean **deleteRecursive**(boolean recursive)

throws `java.io.IOException`

Deletes this `ContentContainer` and all of its `ContentEntry` objects. If the recursive parameter is false, this function behaves in a manner equivalent to `deleteContents()`. This method deletes local `ContentEntry` instances only. If the `#isLocal` method returns false, an exception is thrown. If a `SecurityException` is thrown due to insufficient write access permissions on any entry contained within this `ContentContainer`, this method MAY delete a partial subset of the entries contained within.

**Parameters:**

recursive - if true all entries and their entries are to be deleted.

**Throws:**

`java.lang.SecurityException` - if the caller does not have

`HomeNetPermission("contentmanagement")`, or if the caller does not have write permission on this container or any entries contained in this container.

`java.io.IOException` - if this `ContentContainer` is not local.

**See Also:**

`deleteContents()`, `delete()`

### **addContentEntry**

boolean **addContentEntry**(`ContentEntry` entry)

Adds a `ContentEntry` to this `ContentContainer`. Can only add local `ContentEntry` objects to local `ContentContainer`. If this entry is already has a parent `ContentContainer`, it will be removed from that container.

**Parameters:**

entry - the content entry to be added to this container

**Returns:**

True if the entry was added. Returns false if the `isLocal` method of this `ContentContainer` or the parameter `ContentEntry` returns false.

**Throws:**

`java.lang.IllegalStateException` - if this `ContentContainer` does not have a `parentID` property, i.e., this `ContentContainer` is not added to the CDS.

`java.lang.SecurityException` - if the caller does not have

`HomeNetPermission("contentmanagement")`, or if the caller does not have write permission on this container.

### addContentEntries

boolean **addContentEntries**(ContentEntry[] entries)

Adds ContentEntry objects to this ContentContainer. Can only add local ContentEntry objects to local ContentContainer. If any entry passed to this method already has a parent ContentContainer, it will be removed from that container.

**Parameters:**

entries - the content entries to be added to this container

**Returns:**

True if the entries were added. Returns false if the isLocal method of this ContentContainer or the parameter ContentEntry returns false.

**Throws:**

java.lang.IllegalStateException - if this ContentContainer does not have a parentID property, i.e., this ContentContainer is not added to the CDS.

java.lang.SecurityException - if the caller does not have

HomeNetPermission("contentmanagement"), or if the caller does not have write permission on this container.

### removeContentEntry

boolean **removeContentEntry**(ContentEntry entry)

Removes a ContentEntry from this ContentContainer. Can only add local ContentEntry objects to local ContentContainers. When the ContentEntry parameter is a ContentContainer, all of its ContentEntry objects are removed from the parameter. For entries that are ContentContainer objects, a possible implementation is a recursive traversal where these objects are removed in a bottom-up fashion by calling this method on each one.

**Parameters:**

entry - the content entry to be removed from this container

**Returns:**

True if the entry was removed. Returns false if the isLocal method of this ContentContainer or the parameter ContentEntry is not contained in this container.

**Throws:**

java.lang.IllegalArgumentException - if the ContentEntry parameter is a NetRecordingEntry which contains one or more RecordingContentItems.

java.lang.SecurityException - if the caller does not have

HomeNetPermission("contentmanagement"), or if the caller does not have write permission on this container.

### removeContentEntries

boolean **removeContentEntries**(ContentEntry[] entries)

Removes ContentEntry objects from this ContentContainer. Can only add local ContentEntry objects to local ContentContainer. If any ContentEntry is not contained within this container, this method will return false and no entries will be removed. When the ContentEntry parameter is a ContentContainer, all of its ContentEntry objects are removed from the parameter. For entries that are ContentContainer objects, a possible implementation is a recursive traversal where these objects are removed in a bottom-up fashion by calling removeContentEntry method on each one.

**Parameters:**

entries - the content entries to be removed from this container

**Returns:**

True if the entries were removed. Returns false if the isLocal method of this ContentContainer or if any of the ContentEntry objects are not contained in this container.

**Throws:**

java.lang.IllegalArgumentException - if the parameter includes a NetRecordingEntry which contains one or more RecordingContentItems.

`java.lang.SecurityException` - if the caller does not have `HomeNetPermission("contentmanagement")`, or if the caller does not have write permission on this container.

### **getContentSize**

`long` **getContentSize()**

Gets the size of the `ContentContainer` and all its content including all its contained `ContentContainer` objects. Note that the size may have changed during the call to this method.

**Specified by:**

`getContentSize` in interface `ContentEntry`

**Returns:**

The content size in bytes or -1 if the size is indeterminate.

### **getCreationDate**

`java.util.Date` **getCreationDate()**

Returns the creation date of this `ContentContainer`.

**Specified by:**

`getCreationDate` in interface `ContentEntry`

**Returns:**

The `Date` the content was created or null if the creation date is indeterminate.

### **getExtendedFileAccessPermissions**

`ExtendedFileAccessPermissions` **getExtendedFileAccessPermissions()**

Gets the `ExtendedFileAccessPermissions` of this `ContentContainer`.

**Specified by:**

`getExtendedFileAccessPermissions` in interface `ContentEntry`

**Returns:**

The `ExtendedFileAccessPermission`.

### **getComponentCount**

`int` **getComponentCount()**

Gets the number of `ContentEntry` objects in this `ContentContainer`. Does not include component count of entries within `ContentContainer` objects contained in this `ContentContainer`.

**Returns:**

Number of entries.

### **isEmpty**

`boolean` **isEmpty()**

Returns an empty indication.

**Returns:**

True if this `ContentContainer` does not contain any `ContentEntry` objects, otherwise returns false.

## org.ocap.hn.content Interface ContentEntry

### All Known Subinterfaces:

ContentContainer, ContentItem, NetRecordingEntry, RecordingContentItem

```
public interface ContentEntry
```

This interface represents a basic content entry. Each ContentEntry instance can only be contained in one ContentContainer and the implementation SHALL create a new ContentEntry for equal entries placed in multiple ContentContainer instances.

## Method Summary

boolean	<b>deleteEntry()</b> Deletes this ContentEntry.
long	<b>getContentSize()</b> Gets the size of the content associated with this ContentEntry..
java.util.Date	<b>getCreationDate()</b> Gets the creation date of the content associated with this ContentEntry.
ContentContainer	<b>getEntryParent()</b> Returns the ContentContainer this ContentEntry belongs to.
ExtendedFileAccessPermissions	<b>getExtendedFileAccessPermissions()</b> Gets the file permissions of this ContentEntry, or null if unknown.
java.lang.String	<b>getID()</b> Returns the ID of this ContentEntry.
java.lang.String	<b>getParentID()</b> Returns the ID of ContentContainer this ContentEntry belongs to.
MetadataNode	<b>getRootMetadataNode()</b> Gets the metadata for this ContentEntry.
ContentServerNetModule	<b>getServer()</b> Gets the server where this ContentEntry is located.
boolean	<b>isLocal()</b> Returns true if this content entry is on the local device, false if it is hosted by another device on the network.

## Method Detail

### getID

```
java.lang.String getID()
```

Returns the ID of this ContentEntry. The format of this string ID is implementation and protocol mapping dependent.

**Returns:**

The ID of content entry.

**getServer**

ContentServerNetModule **getServer**()

Gets the server where this ContentEntry is located.

**Returns:**

The server housing this container.

**deleteEntry**

boolean **deleteEntry**()

throws java.io.IOException

Deletes this ContentEntry. This is a local delete only. If the #isLocal method returns false, this method SHALL throw an exception. This method does not delete any content associated with this content entry.

**Returns:**

True if the ContentEntry was deleted, otherwise returns false.

**Throws:**

java.lang.SecurityException - if the calling application does not have write

ExtendedFileAccessPermission for this entry.

java.io.IOException - if the entry is not local.

**getEntryParent**

ContentContainer **getEntryParent**()

throws java.io.IOException

Returns the ContentContainer this ContentEntry belongs to. This method SHALL return null if this ContentEntry represents a root container. If it is determined that this ContentEntry has a parent container, but the implementation does not have sufficient local cached information to construct the ContentContainer, this method SHALL throw an IOException.

**Returns:**

The parent ContentContainer.

**Throws:**

java.io.IOException - if the implementation does not have sufficient local cached information to construct the parent ContentContainer

**getParentID**

java.lang.String **getParentID**()

Returns the ID of ContentContainer this ContentEntry belongs to. This method SHALL return "-1" if this ContentEntry represents a root container. This method SHALL return null if the parent ID is unknown.

**Returns:**

the ID of this entry's parent container

**See Also:**

getID(), getEntryParent()

**getContentSize**

long **getContentSize**()

Gets the size of the content associated with this ContentEntry..

**Returns:**

The content size in bytes or -1 if unknown.

**getCreationDate**

java.util.Date **getCreationDate()**

Gets the creation date of the content associated with this ContentEntry.

**Returns:**

The Date the content was created or null if unknown.

**getExtendedFileAccessPermissions**

ExtendedFileAccessPermissions **getExtendedFileAccessPermissions()**

Gets the file permissions of this ContentEntry, or null if unknown.

**Returns:**

The extended file access permissions of this ContentEntry or null if unknown.

**getRootMetadataNode**

MetadataNode **getRootMetadataNode()**

Gets the metadata for this ContentEntry.

**Returns:**

Root MetadataNode.

**isLocal**

boolean **isLocal()**

Returns true if this content entry is on the local device, false if it is hosted by another device on the network.

**Returns:**

true if the content is local, false otherwise

## org.ocap.hn.content Interface ContentItem

### All Superinterfaces:

ContentEntry

### All Known Subinterfaces:

RecordingContentItem

```
public interface ContentItem
extends ContentEntry
```

This class represents a piece of content. This can be audio, video or still image content. It is not directly linked to any file. This is done via the ContentResources. A ContentItem can have multiple ContentResources. The class itself implements Metadata but the real Metadata is stored in the MetadataNode. That means a call to `getMetadata("foo")` will be translated to `getMetadataRoot().getMetadata("foo")`;

Field Summary	
static java.lang.String	<b>AUDIO_ITEM</b> Represents the base audio content.
static java.lang.String	<b>AUDIO_ITEM_BOOK</b> In addition to being an AUDIO_ITEM content MAY be an audio book.
static java.lang.String	<b>AUDIO_ITEM_BROADCAST</b> In addition to being an AUDIO_ITEM content MAY be broadcast on a radio station.
static java.lang.String	<b>AUDIO_ITEM_TRACK</b> In addition to being an AUDIO_ITEM content MAY be a track such as a song.
static java.lang.String	<b>IMAGE_ITEM</b> Base image item.
static java.lang.String	<b>IMAGE_ITEM_PHOTO</b> In addition to being an IMAGE_ITEM content MAY be a photo.
static java.lang.String	<b>ITEM</b> Represents the base content item.
static java.lang.String	<b>VIDEO_ITEM</b> Represents the base video item.
static java.lang.String	<b>VIDEO_ITEM_BROADCAST</b> In addition to being a VIDEO_ITEM content MAY be a video broadcast.
static java.lang.String	<b>VIDEO_ITEM_MOVIE</b> In addition to being a VIDEO_ITEM content MAY be a movie.
static java.lang.String	<b>VIDEO_ITEM_MUSIC_CLIP</b> In addition to being a VIDEO_ITEM content MAY be a music video clip, e.g.

<b>Method Summary</b>	
boolean	<b>containsResource</b> (ContentResource entry) Checks whether the given ContentResource is part of this ContentItem..
boolean	<b>deleteEntry</b> ( ) Deletes this ContentItem.
java.lang.String	<b>getContentClass</b> ( ) Returns the content class of this content item.
javax.tv.service.Service	<b>getItemService</b> ( ) If this ContentItem is presentable as a JavaTV Service than this method returns a javax.tv.service.Service, or derivative of a Service, e.g.
ContentResource[ ]	<b>getRenderableResources</b> ( ) Gets an array copy of renderable ContentResources which are part of this ContentItem.
ContentResource	<b>getResource</b> (int n) Returns the n <sup>th</sup> ContentResource of this ContentItem.
int	<b>getResourceCount</b> ( ) Returns the number of ContentResources which are associated with this ContentItem.
int	<b>getResourceIndex</b> (ContentResource r) Returns the index of the specified ContentResource or -1 if the ContentResource does not exist in this ContentItem.
ContentResource[ ]	<b>getResources</b> ( ) Gets an array copy of ContentResources which are part of this ContentItem.
java.lang.String	<b>getTitle</b> ( ) Gets the title for this ContentItem, or null if the title is unknown.
boolean	<b>hasAudio</b> ( ) Returns a boolean indicating if this content has audio.
boolean	<b>hasStillImage</b> ( ) Returns a boolean indicating if the ContentItem has a still image.
boolean	<b>hasVideo</b> ( ) Returns a boolean indicating if the ContentItem has video associated with it.
boolean	<b>isRenderable</b> ( ) Checks whether the local device has the capabilities to render this content item.

<b>Methods inherited from interface org.ocap.hn.content.ContentEntry</b>
getContentSize, getCreationDate, getEntryParent, getExtendedFileAccessPermissions, getID, getParentID, getRootMetadataNode, getServer, isLocal

## Field Detail

### ITEM

static final java.lang.String **ITEM**

Represents the base content item.

**See Also:**

Constant Field Values

### AUDIO\_ITEM

static final java.lang.String **AUDIO\_ITEM**

Represents the base audio content.

**See Also:**

Constant Field Values

### AUDIO\_ITEM\_TRACK

static final java.lang.String **AUDIO\_ITEM\_TRACK**

In addition to being an AUDIO\_ITEM content MAY be a track such as a song.

**See Also:**

Constant Field Values

### AUDIO\_ITEM\_BROADCAST

static final java.lang.String **AUDIO\_ITEM\_BROADCAST**

In addition to being an AUDIO\_ITEM content MAY be broadcast on a radio station.

**See Also:**

Constant Field Values

### AUDIO\_ITEM\_BOOK

static final java.lang.String **AUDIO\_ITEM\_BOOK**

In addition to being an AUDIO\_ITEM content MAY be an audio book.

**See Also:**

Constant Field Values

### VIDEO\_ITEM

static final java.lang.String **VIDEO\_ITEM**

Represents the base video item.

**See Also:**

Constant Field Values

### VIDEO\_ITEM\_MOVIE

static final java.lang.String **VIDEO\_ITEM\_MOVIE**

In addition to being a VIDEO\_ITEM content MAY be a movie.

**See Also:**

Constant Field Values

### VIDEO\_ITEM\_BROADCAST

static final java.lang.String **VIDEO\_ITEM\_BROADCAST**

In addition to being a VIDEO\_ITEM content MAY be a video broadcast.

**See Also:**

Constant Field Values

### VIDEO\_ITEM\_MUSIC\_CLIP

```
static final java.lang.String VIDEO_ITEM_MUSIC_CLIP
```

In addition to being a VIDEO\_ITEM content MAY be a music video clip, e.g. music video.

**See Also:**

Constant Field Values

### IMAGE\_ITEM

```
static final java.lang.String IMAGE_ITEM
```

Base image item.

**See Also:**

Constant Field Values

### IMAGE\_ITEM\_PHOTO

```
static final java.lang.String IMAGE_ITEM_PHOTO
```

In addition to being an IMAGE\_ITEM content MAY be a photo.

**See Also:**

Constant Field Values

## Method Detail

### hasAudio

```
boolean hasAudio()
```

Returns a boolean indicating if this content has audio.

**Returns:**

True if the content type has audio, otherwise returns false.

### hasVideo

```
boolean hasVideo()
```

Returns a boolean indicating if the ContentItem has video associated with it.

**Returns:**

True if the ContentItem contains video, otherwise returns false.

### hasStillImage

```
boolean hasStillImage()
```

Returns a boolean indicating if the ContentItem has a still image.

**Returns:**

True if the ContentItem has a still image, otherwise returns false.

### getItemService

```
javax.tv.service.Service getItemService()
```

If this ContentItem is presentable as a JavaTV Service than this method returns a javax.tv.service.Service, or derivative of a Service, e.g. RecordedService, which can be used to play this ContentItem. If the

ContentItem is not local the returned Service SHALL be a RemoteService. If the content associated with item has been deleted or is no longer accessible, this method SHALL return null.

**Returns:**

A JavaTV service if this content is presentable as a Service, null otherwise.

**getContentClass**

```
java.lang.String getContentClass()
```

Returns the content class of this content item.

**Returns:**

The content class of this item.

**See Also:**

AUDIO\_ITEM, AUDIO\_ITEM\_BOOK, AUDIO\_ITEM\_BROADCAST, AUDIO\_ITEM\_TRACK,  
IMAGE\_ITEM, VIDEO\_ITEM, VIDEO\_ITEM\_BROADCAST, VIDEO\_ITEM\_MOVIE,  
VIDEO\_ITEM\_MUSIC\_CLIP, IMAGE\_ITEM\_PHOTO

**getTitle**

```
java.lang.String getTitle()
```

Gets the title for this ContentItem, or null if the title is unknown.

**Returns:**

the String title for this item, or null if unknown.

**deleteEntry**

```
boolean deleteEntry()
```

throws java.io.IOException

Deletes this ContentItem. Calls the ContentResource.delete() method on each ContentResource contained in this ContentItem. Deletes a local ContentItem only. If the #isLocal method returns false an exception is thrown.

Note: this overrides the definition of ContentEntry. If an application calls the ContentEntry.deleteEntry method on an object that is an instance of ContentItem, the implementation SHALL delete the ContentItem as defined by this method.

The ContentItem is not valid anymore after this call.

**Specified by:**

deleteEntry in interface ContentEntry

**Returns:**

True if the ContentEntry was deleted, otherwise returns false.

**Throws:**

java.lang.SecurityException - if the application does not have write  
ExtendedFileAccessPermission.

java.io.IOException - if the ContentItem is not local.

**getResourceCount**

```
int getResourceCount()
```

Returns the number of ContentResources which are associated with this ContentItem.

**Returns:**

number of ContentResources.

**getResource**

```
ContentResource getResource(int n)
```

Returns the n<sup>th</sup> ContentResource of this ContentItem.

**Parameters:**

n - the index of the ContentResource

**Returns:**

the n<sup>th</sup>

**Throws:**

java.lang.ArrayIndexOutOfBoundsException - if the n<sup>th</sup> value does not exist.

**getResourceIndex**

int **getResourceIndex**(ContentResource r)

Returns the index of the specified ContentResource or -1 if the ContentResource does not exist in this ContentItem.

**Parameters:**

r - The ContentResource to check for.

**Returns:**

The index of the ContentResource or -1 if it doesn't exist in this ContentItem.

**containsResource**

boolean **containsResource**(ContentResource entry)

Checks whether the given ContentResource is part of this ContentItem..

**Parameters:**

entry - The ContentResource to check for.

**Returns:**

True if the ContentResource is part of this ContentItem, otherwise returns false.

**getResources**

ContentResource[] **getResources**()

Gets an array copy of ContentResources which are part of this ContentItem.

**Returns:**

Array of ContentResources.

**getRenderableResources**

ContentResource[] **getRenderableResources**()

Gets an array copy of renderable ContentResources which are part of this ContentItem.

**Returns:**

Array of ContentResources contained in this ContentItem for which ContentResource.isRenderable() returns true.

**isRenderable**

boolean **isRenderable**()

Checks whether the local device has the capabilities to render this content item. This includes the ability to negotiate media protocol with the host device, the ability of the local device to render this content item's media format, and sufficient access permissions for the calling application. This method will return true if any of the ContentResources contained in this ContentItem are renderable. This call does not consider immediate availability of resources required for presentation of this content.

**Returns:**

true if this content is renderable on the local device, false otherwise.

## org.ocap.hn.content Interface ContentResource

### All Known Subinterfaces:

AudioResource, StreamableContentResource, VideoResource

```
public interface ContentResource
```

Abstract class representing a media stream/file. Subclasses of this class can implement AudioResource and/or VideoResource depending on whether the content represents audio and/or video.

## Field Summary

static java.lang.String	<b>UNKNOWN_MIME_TYPE</b> Constant for an unknown MIME type.
-------------------------	--

## Method Summary

boolean	<b>delete()</b> Deletes the binary representation of this ContentResource and the ContentResource is removed from any containing ContentEntry.
java.lang.String	<b>getContentFormat()</b> Returns the content format.
ContentItem	<b>getContentItem()</b> Gets the ContentItem this resource belongs to.
long	<b>getContentSize()</b> Gets the size of the content in bytes or -1 if not known.
java.util.Date	<b>getCreationDate()</b> Gets the creation date of the content or NULL if not known.
ExtendedFileAccessPermissions	<b>getExtendedFileAccessPermissions()</b> Returns the file permissions of a ContentResource.
javax.tv.locator.Locator	<b>getLocator()</b> Gets an OcapLocator to the content associated with this ContentResource if the content can be located with that Locator type, otherwise returns an implementation specific Locator to the content.
java.lang.String	<b>getNetwork()</b> Returns the network on which the content is available.
java.lang.String	<b>getProtocol()</b> Returns the protocol which can be used to retrieve the content.
java.lang.Object	<b>getResourceProperty(java.lang.String key)</b> Returns properties of the resource.
boolean	<b>isRenderable()</b> Checks whether the local device has the capabilities to render this content resource.

## Field Detail

### UNKNOWN\_MIME\_TYPE

static final java.lang.String **UNKNOWN\_MIME\_TYPE**

Constant for an unknown MIME type.

**See Also:**

Constant Field Values

## Method Detail

### delete

boolean **delete()**

throws java.io.IOException

Deletes the binary representation of this ContentResource and the ContentResource is removed from any containing ContentEntry. The ContentResource is not valid anymore after this call. This method deletes a local ContentResource only. If the #isLocal method returns false an exception is thrown. Does not delete the content associated with this ContentResource; see #getLocator.

**Returns:**

True if this ContentResource was deleted, otherwise returns false.

**Throws:**

java.lang.SecurityException - if the application does not have a write ExtendedFileAccessPermission.

java.io.IOException - if the ContentResource is not local.

### getContentItem

ContentItem **getContentItem()**

Gets the ContentItem this resource belongs to.

**Returns:**

The ContentItem parent of this resource or null if this ContentResource is an independent ContentEntry.

### getContentSize

long **getContentSize()**

Gets the size of the content in bytes or -1 if not known.

**Returns:**

the content size in bytes

### getCreationDate

java.util.Date **getCreationDate()**

Gets the creation date of the content or NULL if not known.

**Returns:**

The Date the content was created.

### getLocator

javax.tv.locator.Locator **getLocator()**

Gets an OcapLocator to the content associated with this ContentResource if the content can be located with that Locator type, otherwise returns an implementation specific Locator to the content.

**Returns:**

Locator to the content associated with this entry.

**getExtendedFileAccessPermissions**

ExtendedFileAccessPermissions **getExtendedFileAccessPermissions()**

Returns the file permissions of a ContentResource.

**Returns:**

the extended file access permissions of the ContentEntry.

**getProtocol**

java.lang.String **getProtocol()**

Returns the protocol which can be used to retrieve the content. The returned String can be a wild card "\*".

Possible protocols are

- "http-get"
- "rtsp-rtp-udp"
- "internal"
- "iec61883"
- Registered ICANN domain name of vendor

**Returns:**

String representation of the protocol

**getNetwork**

java.lang.String **getNetwork()**

Returns the network on which the content is available. The returned String can be a wild card "\*".

<protocol>	<network>
"http-get"	"*"
"rtsp-rtp-udp"	"*"
"internal"	IP address of the device hosting the Connection manager
"iec61883"	GUID of the 1394 bus Isochronous Resource Manager
ICANN domain	Vendor-defined, may be "*"

**Returns:**

String describing the network on which the resource is available.

**getContentFormat**

java.lang.String **getContentFormat()**

Returns the content format. The returned String can be a wild card "\*".

<protocol>	<network>
"http-get"	MIME-type
"rtsp-rtp-udp"	Name of RTP payload type
"internal"	Vendor-defined, may be "*"
"iec61883"	Name standardised by IEC61883
ICANN	Vendor-defined, may be "*"

domain

**Returns:**

String describing the content format.

### **getResourceProperty**

java.lang.Object **getResourceProperty**(java.lang.String key)

Returns properties of the resource. There is a set of defined properties which can be accessed via the methods in `AudioResource` and `VideoResource`. This method allows for custom or new properties.

**Parameters:**

key - The key of the property.

**Returns:**

The value of the property, or null if the key parameter does not match any property.

### **isRenderable**

boolean **isRenderable**()

Checks whether the local device has the capabilities to render this content resource. This includes the ability to negotiate media protocol with the host device, the ability of the local device to render this content item's media format, and sufficient access permissions for the calling application. This call does not consider immediate availability of resources required for presentation of this content.

**Returns:**

true if this content is renderable on the local device, false otherwise.

## org.ocap.hn.content Class DatabaseException

```
java.lang.Object
├ java.lang.Throwable
│   └ java.lang.Exception
│       └ org.ocap.hn.content.DatabaseException
```

### All Implemented Interfaces:

```
java.io.Serializable
```

```
public class DatabaseException
extends java.lang.Exception
```

Exception that is thrown when a database error occurs

### See Also:

Serialized Form

## Field Summary

static int	<b>FIELD_IS_EMPTY</b>
static int	<b>FIELD_IS_WRONG_FORMAT</b>
static int	<b>FIELD_NAME DOES NOT EXIST</b>
static int	<b>GENERAL_ERROR</b>
static int	<b>INVALID_PARAMETER_SPECIFIED</b>
static int	<b>QUERY_IS_INVALID</b>
static int	<b>REMOTE_QUERY_IS_INVALID</b>
static int	<b>UNABLE_TO_LOCATE_SERVICE</b>

## Constructor Summary

```
DatabaseException(java.lang.String sMessage, int iNumber)
```

## Method Summary

int	<b>getExceptionNumber()</b> Returns the unique exception code related to the database.
-----	---

**Methods inherited from class java.lang.Throwable**

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

**FIELD\_NAME\_DOES\_NOT\_EXIST**

public static final int **FIELD\_NAME\_DOES\_NOT\_EXIST**

**See Also:**

Constant Field Values

**FIELD\_IS\_EMPTY**

public static final int **FIELD\_IS\_EMPTY**

**See Also:**

Constant Field Values

**FIELD\_IS\_WRONG\_FORMAT**

public static final int **FIELD\_IS\_WRONG\_FORMAT**

**See Also:**

Constant Field Values

**QUERY\_IS\_INVALID**

public static final int **QUERY\_IS\_INVALID**

**See Also:**

Constant Field Values

**INVALID\_PARAMETER\_SPECIFIED**

public static final int **INVALID\_PARAMETER\_SPECIFIED**

**See Also:**

Constant Field Values

**UNABLE\_TO\_LOCATE\_SERVICE**

public static final int **UNABLE\_TO\_LOCATE\_SERVICE**

**See Also:**

Constant Field Values

**REMOTE\_QUERY\_IS\_INVALID**

public static final int **REMOTE\_QUERY\_IS\_INVALID**

**See Also:**

Constant Field Values

**GENERAL\_ERROR**

```
public static final int GENERAL_ERROR
```

**See Also:**

Constant Field Values

## Constructor Detail

**DatabaseException**

```
public DatabaseException(java.lang.String sMessage,  
                          int iNumber)
```

## Method Detail

**getExceptionNumber**

```
public int getExceptionNumber()
```

Returns the unique exception code related to the database.

**Returns:**

the integer value with an exception code

**org.ocap.hn.content**  
**Interface IOStatus**

```
public interface IOStatus
```

This interface represents the ability to detect whether any asset represented by an object or its children is in use on the home network and hence the object should not be deleted.

ContentResource and ContentEntry objects representing local assets on a server SHALL implement IOStatus.

## Method Summary

boolean	<b>isInUse()</b>	Returns an indication of whether any asset within this object is in use on the home network.
---------	------------------	--

## Method Detail

**isInUse**

```
boolean isInUse()
```

Returns an indication of whether any asset within this object is in use on the home network. "In Use" is indicated if there is an active network transport protocol session (for example HTTP, RTSP) to the asset.

For objects which logically contain other objects, recursively iterates through all logical children of this object. For ContentContainer objects, recurses through all ContentEntry objects they contain. For NetRecordingEntry objects, iterates through all RecordingContentItem objects they contain.

**Returns:**

True if there is an active network transport protocol session to any asset that this ContentResource, ContentEntry, or any children of the ContentEntry contain, otherwise false.

## org.ocap.hn.content Class MetadataIdentifiers

```
java.lang.Object
└─org.ocap.hn.content.MetadataIdentifiers
```

```
public abstract class MetadataIdentifiers
extends java.lang.Object
```

This abstract class represents access to standardized metadata identifiers. Each identifier, e.g. "title", can be used to search for corresponding metadata in a ContentList. The set of identifiers returned by the #getIdentifiers method SHALL contain the PROPRIETARY\_DATA identifier and MAY contain identifiers defined in other OCAP HN profiles, e.g. UPnP.

### Field Summary

static java.lang.String	<b>PROPRIETARY_DATA</b> This identifies proprietary data.
-------------------------	--

### Constructor Summary

<b>MetadataIdentifiers()</b>
------------------------------

### Method Summary

static boolean	<b>contains</b> (java.lang.String identifier) Indicates if the parameter identifier is contained within the set of supported identifiers.
static java.lang.String[]	<b>getIdentifiers</b> () Gets all metadata identifiers for all HN profiles supported by this Host device.
static int	<b>getNumberOfIdentifiers</b> () Gets the number of identifiers in the set of supported identifiers returned by the #getIdentifiers method.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### PROPRIETARY\_DATA

```
public static final java.lang.String PROPRIETARY_DATA
```

This identifies proprietary data. The Object returned when using this as a Metadata identifier is defined by the application creating the metadata.

The value of this field is an OCAP defined string "ocap:proprietaryData". If the proprietary data is an array of bytes the data should be transported as a base 64 String.

**See Also:**  
Constant Field Values

## Constructor Detail

### MetadataIdentifiers

```
public MetadataIdentifiers()
```

## Method Detail

### getIdentifiers

```
public static java.lang.String[] getIdentifiers()
```

Gets all metadata identifiers for all HN profiles supported by this Host device.

**Returns:**  
Array of Metadata identifiers.

### getNumberOfIdentifiers

```
public static int getNumberOfIdentifiers()
```

Gets the number of identifiers in the set of supported identifiers returned by the #getIdentifiers method.

**Returns:**  
Number of supported metadata identifiers.

### contains

```
public static boolean contains(java.lang.String identifier)
```

Indicates if the parameter identifier is contained within the set of supported identifiers.

**Parameters:**  
identifier - Name of the identifier to search for.

**Returns:**  
True if the identifier is supported, otherwise returns false.

**org.ocap.hn.content**  
**Class MetadataNode**

```
java.lang.Object
└─ org.ocap.hn.content.MetadataNode

public abstract class MetadataNode
extends java.lang.Object
```

Base class for all Metadata. Metadata can be a normal java.lang.String, any serializable Object, another MetadataNode for tree like meta data structures or other Objects declared as supported by this specification. One example is given below:

```
RootNode
|
--- TITLE - String("Best Movie Ever")
|
--- CREW - MetadataNode
|
|   --- MAIN_ACTOR - String("Joe Sixpack")
|   |
|   --- MAIN_ACTOR2 - String("Doris Dosenkohl")
|
--- SYNOPSIS - String("Don't know - I fell asleep after 5 seconds")
```

It is possible to get Metadata from other MetadataNodes directly by concatenating the different identifiers using # to separate them.

e.g. getMetadata("CREW#MAIN\_ACTOR") would return "Joe Sixpack"

The MetadataNode represents the current snapshot of metadata associated with a network entity as cached on the local device. This may or may not reflect an accurate or complete view of the metadata that exists on the network. It is the responsibility of the application to explicitly update metadata using the home network APIs (e.g. ContentServerNetModule.requestSearchEntries(String, String, int, int, String, String, org.ocap.hn.NetActionHandler)

**Constructor Summary**

**MetadataNode()**

**Method Summary**

abstract void	<b>addMetadata</b> (java.lang.String key, java.lang.Object value) Adds a new metadata entry to this MetadataNode or modifies an existing metadata entry in this MetadataNode.
abstract void	<b>addMetadata</b> (java.lang.String key, java.lang.Object value, ExtendedFileAccessPermissions efap) Adds a new metadata entry to this MetadataNode or modifies an existing metadata entry in this MetadataNode.

<b>Method Summary</b>	
abstract void	<b>addNamespace</b> ( java.lang.String namespace, java.lang.String URI) Adds a namespace definition to this MetadataNode.
static MetadataNode	<b>createMetadataNode</b> ( java.lang.String key) Creates an instance of a MetadataNode object.
abstract ExtendedFileAccessPermissions	<b>getExtendedFileAccessPermissions</b> ( java.lang.String key) Gets the extended file access permissions for a property.
abstract java.lang.String	<b>getKey</b> ( ) Gets the key string which can be utilized to retrieve this MetadataNode from this node's parent.
abstract java.lang.String[]	<b>getKeys</b> ( ) Gets the set of keys for all metadata contained within this MetadataNode.
abstract java.util.Enumeration	<b>getMetadata</b> ( ) Gets an Enumeration of all metadata objects in this node.
abstract java.lang.Object	<b>getMetadata</b> ( java.lang.String key) Returns the Metadata for the specified key.
abstract MetadataNode	<b>getParentNode</b> ( ) Gets the parent node of this MetadataNode.

<b>Methods inherited from class java.lang.Object</b>
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### MetadataNode

public **MetadataNode**( )

## Method Detail

### createMetadataNode

public static MetadataNode **createMetadataNode**( java.lang.String key)  
Creates an instance of a MetadataNode object. When added to another MetadataNode, this node can be located using the key specified in the constructor.

**Parameters:**

key - the key for locating this MetadataNode

**Returns:**

the newly created MetadataNode

### getMetadata

public abstract java.lang.Object **getMetadata**( java.lang.String key)

Returns the Metadata for the specified key. The returned object can be a normal String (e.g. for title etc), a `java.awt.Image`, a `Vector` containing a list of elements, another `MetadataNode` for tree like metadata structures or other tbd. Objects. In order to query a Metadata in a hierarchy query strings like `IDENT1#IDENT2#IDENT3` are possible where `IDENT1` is a `MetadataNode` in this node, `IDENT2` a `MetadataNode` in `IDENT1` and `IDENT3` an object in `IDENT2`. Performs a local cache search only. This method only returns locally cached metadata. This method SHALL not cause network activity.

**Parameters:**

`key` - The key to search for.

**Returns:**

The value associated with the key or null if no match found.

**Throws:**

`java.lang.SecurityException` - if the calling application does not have extended file access permission to read from the property represented by the key parameter.

**addMetadata**

```
public abstract void addMetadata(java.lang.String key,
                                java.lang.Object value)
```

Adds a new metadata entry to this `MetadataNode` or modifies an existing metadata entry in this `MetadataNode`. The value Object to be added to this metadata node MUST be of a type for which a known network mapping exists. All objects implementing the `Serializable` interface are considered to have a known network mapping. In addition, the following classes have a known network mapping:

```
org.dvb.application.AppID
org.ocap.storage.ExtendedFileAccessPermissions
org.ocap.hn.content.ContentContainer
org.ocap.hn.content.MetadataNode
```

If the Object to be added is an instance of `MetadataNode`, all Objects contained within that `MetadataNode` SHALL be recursively added to this `MetadataNode` as a result of this operation.

Invocation of this method SHALL replace any values already present for the given `key` parameter. If a value Object is already associated with the key string passed into this method, the value passed in by this method SHALL replace the existing value. If a value of null is passed into this method, any existing value associated with the key passed into this method SHALL be removed. Keys MAY contain a namespace as part of their definition. Namespaces SHALL appear in keys as a colon separated prefix in the key string (e.g. "`namespace:key`"). Vendor specific namespaces MUST be added to this `MetadataNode` using the `addNameSpace()` method prior to usage in this method, or be a valid namespace in any containing parent `MetadataNode`. Each property added by this method whether new or pre-existing SHALL be given an `ExtendedFileAccessPermissions` that matches that of the parent `ContentEntry`.

**Parameters:**

`key` - The key e.g. "TITLE". When the value type is `MetadataNode` the implementation SHALL overwrite the key set by the `createMetadataNode` method with the value of this parameter.

`value` - The value associated with the key, e.g. "When Harry Meets Sally" or a more complex Object like another `MetadataNode`.

**Throws:**

`java.lang.IllegalArgumentException` - if value is not a known mappable object type, or if the key parameter contains an unknown vendor defined namespace .

`java.lang.SecurityException` - if the property identified by the key parameter does not exist and the calling application does not have sufficient file access permissions to write to the `ContentEntry` containing this `MetadataNode`.

`java.lang.SecurityException` - if the property identified by the key parameter does exist and the calling application does not have sufficient file access permissions to write to the property.

**See Also:**

```
addNameSpace(String namespace, String URI)
```

**addMetadata**

```
public abstract void addMetadata(java.lang.String key,  
                                 java.lang.Object value,  
                                 ExtendedFileAccessPermissions efap)
```

Adds a new metadata entry to this MetadataNode or modifies an existing metadata entry in this MetadataNode. The value Object to be added to this metadata node **MUST** be of a type for which a known network mapping exists. All objects implementing the Serializable interface are considered to have a known network mapping. In addition, the following classes have a known network mapping:

```
org.dvb.application.AppID  
org.ocap.storage.ExtendedFileAccessPermissions  
org.ocap.hn.content.ContentContainer  
org.ocap.hn.content.MetadataNode
```

If the Object to be added is an instance of MetadataNode, all Objects contained within that MetadataNode **SHALL** be recursively added to this MetadataNode as a result of this operation.

Invocation of this method **SHALL** replace any values already present for the given *key* parameter. If a value Object is already associated with the key string passed into this method, the value passed in by this method **SHALL** replace the existing value. If a value of null is passed into this method, any existing value associated with the key passed into this method **SHALL** be removed. Keys **MAY** contain a namespace as part of their definition. Namespaces **SHALL** appear in keys as a colon separated prefix in the key string (e.g. "*namespace:key*"). Vendor specific namespaces **MUST** be added to this MetadataNode using the addNameSpace() method prior to usage in this method, or be a valid namespace in any containing parent MetadataNode. Each property added by this method whether new or pre-existing **SHALL** be given an ExtendedFileAccessPermissions that matches the efap parameter.

**Parameters:**

key - The key e.g. "TITLE". When the value type is MetadataNode the implementation **SHALL** overwrite the key set by the createMetadataNode method with the value of this parameter.

value - The value associated with the key, e.g. "When Harry Meets Sally" or a more complex Object like another MetadataNode.

efap - ExtendedFileAccessPermissions for the property or properties added or modified by this method.

**Throws:**

java.lang.IllegalArgumentException - if value is not a known mappable object type, or if the key parameter contains an unknown vendor defined namespace, or if the efap parameter is null.

java.lang.SecurityException - if the property identified by the key parameter does not exist and the calling application does not have sufficient file access permissions to write to the ContentEntry containing this MetadataNode.

java.lang.SecurityException - if the property identified by the key parameter does exist and the calling application does not have sufficient file access permissions to write to the property.

**See Also:**

```
addNameSpace(String namespace, String URI)
```

**addNameSpace**

```
public abstract void addNameSpace(java.lang.String namespace,  
                                   java.lang.String URI)
```

Adds a namespace definition to this MetadataNode. Once added, this namespace **SHALL** be valid as a metadata key qualifier for this node and any children and sub-children of this MetadataNode.

**Parameters:**

namespace - String identifier of the namespace to be added

URI - vendor specific String based universal resource indicator pointing to the definition of this namespace.

**See Also:**

```
addMetadata(String key, Object value)
```

**getMetadata**

```
public abstract java.util.Enumeration getMetadata()
```

Gets an Enumeration of all metadata objects in this node. The enumeration SHALL contain the top-level MetadataNode objects which MAY contain other MetadataNode objects. This method only returns locally cached metadata. This method SHALL not cause network activity.

**Returns:**

Enumeration of top-level MetadataNode objects.

**getParentNode**

```
public abstract MetadataNode getParentNode()
```

Gets the parent node of this MetadataNode.

**Returns:**

Parent of this node.

**getKey**

```
public abstract java.lang.String getKey()
```

Gets the key string which can be utilized to retrieve this MetadataNode from this node's parent. If this MetadataNode is a root MetadataNode, the method SHALL return null.

**Returns:**

the key string associated with this MetadataNode

**getKeys**

```
public abstract java.lang.String[] getKeys()
```

Gets the set of keys for all metadata contained within this MetadataNode.

**Returns:**

an array of Strings representing metadata contained within this MetadataNode

**getExtendedFileAccessPermissions**

```
public abstract ExtendedFileAccessPermissions
```

```
getExtendedFileAccessPermissions(java.lang.String key)
```

Gets the extended file access permissions for a property.

**Parameters:**

key - The key value representing the property to get the permissions for.

**Returns:**

The extended file access permissions of the property or null if the key does not match a known property.

## org.ocap.hn.content Interface StreamableContentResource

**All Superinterfaces:**  
ContentResource

```
public interface StreamableContentResource
extends ContentResource
```

Abstract class representing content that can be streamed, e.g., MPEG file.

### Field Summary

Fields inherited from interface org.ocap.hn.content.ContentResource

UNKNOWN\_MIME\_TYPE

### Method Summary

long	<b>getBitrate()</b> Gets the Bitrate this content is encoded with or -1 if not known.
javax.media.Time	<b>getDuration()</b> Gets the duration of this content.

Methods inherited from interface org.ocap.hn.content.ContentResource

delete, getContentFormat, getContentItem, getContentSize, getCreationDate, getExtendedFileAccessPermissions, getLocator, getNetwork, getProtocol, getResourceProperty, isRenderable

### Method Detail

#### getDuration

```
javax.media.Time getDuration()
```

Gets the duration of this content.  
**Returns:**  
the duration of the content.

#### getBitrate

```
long getBitrate()
```

Gets the Bitrate this content is encoded with or -1 if not known.  
**Returns:**  
the bit rate of the content in bytes per second or -1 if not known.

## org.ocap.hn.content Interface VideoResource

**All Superinterfaces:**  
ContentResource

```
public interface VideoResource
extends ContentResource
```

ContentResource to identify that a content item contains video/still image material.

### Field Summary

Fields inherited from interface org.ocap.hn.content.ContentResource

UNKNOWN\_MIME\_TYPE

### Method Summary

int	<b>getColorDepth()</b> Returns the color depth (in bits) of the video/still image.
java.awt.Dimension	<b>getResolution()</b> Returns the resolution of the video/still image.

Methods inherited from interface org.ocap.hn.content.ContentResource

delete, getContentFormat, getContentItem, getContentSize, getCreationDate, getExtendedFileAccessPermissions, getLocator, getNetwork, getProtocol, getResourceProperty, isRenderable

### Method Detail

#### getResolution

```
java.awt.Dimension getResolution()
```

Returns the resolution of the video/still image.  
**Returns:**  
the resolution of the video/still image

#### getColorDepth

```
int getColorDepth()
```

Returns the color depth (in bits) of the video/still image.  
**Returns:**  
the color depth (in bits) of the video/still image.

## Annex C Content Navigation API

### Package org.ocap.hn.content.navigation

#### Interface Summary

<b>ContentList</b>	This interface represents a list of filtered ContentEntry objects.
--------------------	--

#### Class Summary

<b>ContentDatabaseFilter</b>	This class represent a filtering criteria to be applied while creating a ContentList.
------------------------------	---

<b>DatabaseQuery</b>	DatabaseQuery class.
----------------------	----------------------

<b>DeviceFilter</b>	This class represents a filtering criteria based on a particular device.
---------------------	--

## org.ocap.hn.content.navigation Class ContentDatabaseFilter

```
java.lang.Object
└─ org.ocap.hn.content.navigation.ContentDatabaseFilter
Direct Known Subclasses:
    DatabaseQuery, DeviceFilter
```

```
public abstract class ContentDatabaseFilter
extends java.lang.Object
```

This class represent a filtering criteria to be applied while creating a ContentList.

### Constructor Summary

protected	<b>ContentDatabaseFilter</b> ()
-----------	---------------------------------

### Method Summary

abstract	<b>accept</b> (ContentEntry entry)
boolean	This method is called for every ContentEntry in the database/list this filter is applied to.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### ContentDatabaseFilter

```
protected ContentDatabaseFilter()
```

### Method Detail

#### accept

```
public abstract boolean accept(ContentEntry entry)
```

This method is called for every ContentEntry in the database/list this filter is applied to. Implementations should return true if the specified ContentItem should be in the filtered list. If the ContentItem should not be listed in the new list false should be returned.

#### Parameters:

entry - the ContentEntry to filter

#### Returns:

true if the ContentEntry should be in the filtered ContentList, false otherwise.

## org.ocap.hn.content.navigation Interface ContentList

### All Superinterfaces:

java.util.Enumeration

```
public interface ContentList
extends java.util.Enumeration
```

This interface represents a list of filtered ContentEntry objects. A ContentList may contain a complete or partial subset of entries resulting from an application requested filter, browse or search.

## Method Summary

ContentList	<b>filterContentList</b> (ContentDatabaseFilter filter) Filters the ContentList.
ContentEntry	<b>find</b> (java.lang.String[] keys, java.lang.Object[] values) Finds the first ContentEntry which matches the search.
ContentEntry	<b>find</b> (java.lang.String key, java.lang.Object value) Finds the first <a href="#">ContentEntry</a> which identifier for the key 'key' equals the given object obj.
ContentList	<b>findAll</b> (java.lang.String[] keys, java.lang.Object[] values) Finds all ContentEntry objects which match the search.
java.lang.String	<b>getSortOrder</b> () Gets the sort order set by the #setSortOrder method.
void	<b>setSortOrder</b> (java.lang.String sortOrder) Sets the metadata sort order of the items in this list based on metadata key identifiers using signed property values.
int	<b>size</b> () Gets the number of ContentEntry objects in this ContentList.
int	<b>totalMatches</b> () Gets to total number of ContentEntry matches in the filter, browse or search operation that generated this ContentList.

### Methods inherited from interface java.util.Enumeration

hasMoreElements, nextElement

## Method Detail

### size

```
int size()
```

Gets the number of ContentEntry objects in this ContentList.

#### Returns:

Number of entries in this list. Returns 0 if the list is empty.

**totalMatches**

```
int totalMatches()
```

Gets to total number of ContentEntry matches in the filter, browse or search operation that generated this ContentList. This value SHALL be greater than or equal to the value returned from the size() method of this ContentList. This value SHALL be greater than the value returned from the size() method of this ContentList if the *requestedCount* parameter of the originating content entry request was less than the total entry matches of the requesting operation. See ContentServerNetModule.

**Returns:**

the total number of ContentEntry matches from the originating content entry request

**setSortOrder**

```
void setSortOrder(java.lang.String sortOrder)
```

Sets the metadata sort order of the items in this list based on metadata key identifiers using signed property values. The sortOrder parameter of this method is a string containing the properties and sort modifiers to be used to sort the resulting ContentList. The format of the string containing the sort criteria shall follow the format defined in UPnP Content Directory Service 3.0 specification section 2.3.16:

A\_ARG\_TYPE\_SortCriteria.

**Parameters:**

sortOrder - a String representing the sortOrder for this ContentList.

**getSortOrder**

```
java.lang.String getSortOrder()
```

Gets the sort order set by the #setSortOrder method.

**Returns:**

The array of sort keys, or null if the setPreferredSortOrder method has not been called for this list.

**find**

```
ContentEntry find(java.lang.String key,
                 java.lang.Object value)
```

Finds the first ContentEntry which identifier for the key 'key' equals the given object obj. For instance, if key == "Title" then obj represents the title, e.g. "Best movie ever" and this method will return the first ContentEntry in the list than contains a match for the (key, value) pair.

**Parameters:**

key - The identifier key.

value - The object to compare to

**Returns:**

The first matched ContentEntry, or null if no match found.

**find**

```
ContentEntry find(java.lang.String[] keys,
                 java.lang.Object[] values)
```

Finds the first ContentEntry which matches the search. The keys and values parameters are parallel arrays. For example, if keys[0] == "TITLE" and values[0] == "Best movie ever", the implementation SHALL match the first ContentEntry in the list where the metadata contains that (key, value) pair, as well as matches any other entries in the parameter arrays.

**Parameters:**

keys - Array of identifier keys.

values - Array of values.

**Returns:**

The first matching ContentEntry found, or null if no match. If the parameter arrays are not the same length this method returns null.

### findAll

```
ContentList findAll(java.lang.String[] keys,  
                   java.lang.Object[] values)
```

Finds all ContentEntry objects which match the search. Same as the #find(String[], Object[]) method except all matches are returned instead of just the first match.

**Parameters:**

keys - Array of identifier keys.

values - Array of values.

**Returns:**

A ContentList containing all matches, or null if no matches were found.

### filterContentList

```
ContentList filterContentList(ContentDatabaseFilter filter)  
                               throws DatabaseException
```

Filters the ContentList. The returned ContentList is a new ContentList only containing ContentItems on which ContentDatabaseFilter.accept returned true.

**Parameters:**

filter - the ContentDatabaseFilter

**Returns:**

newly created ContentList containing only the filtered ContentItems.

**Throws:**

DatabaseException; - see DatabaseException for exception reasons.

DatabaseException

## org.ocap.hn.content.navigation Class DatabaseQuery

```
java.lang.Object
├─ org.ocap.hn.content.navigation.ContentDatabaseFilter
│   └─ org.ocap.hn.content.navigation.DatabaseQuery
```

```
public abstract class DatabaseQuery
extends ContentDatabaseFilter
```

DatabaseQuery class. This class represents a query of the metadata database. Methods are provided so that complex AND, OR, NOT expressions can be created.

Note that this object is immutable; the `and()`, `or()` and `negate()` methods do not affect this object but return references to a new query.

For example:

```
DatabaseQuery qa = DatabaseQuery.newInstance("Genre",
DatabaseQuery.EQUALS, "3.4.11");
DatabaseQuery qb = DatabaseQuery.newInstance("SpokenLanguage",
DatabaseQuery.EQUALS, "fr-CA");
DatabaseQuery qc = qb.and(qa.negate());
```

So the final statement has no effect on the state of `qa` or `qb` objects, which still represent non-negated, root predicates.

Field Summary	
static int	<b>CONTAINS</b> Operator to check for a String being contained within the field The comparison is case insensitive, non whole word.
static int	<b>EQUALS</b> Used to specify a test for equality.
static int	<b>EXISTS</b> Operator to check to see if a field exists
static int	<b>GREATER_THAN</b> Operator to specify greater than .
static int	<b>GREATER_THAN_OR_EQUALS</b> Operator to specify greater than or equals.
static int	<b>LESS_THAN</b> Operator to specify less than .
static int	<b>LESS_THAN_OR_EQUALS</b> Operator to specify less than or equals.
static int	<b>NOT_EQUALS</b> Operator to check for in-equality.

## Constructor Summary

`DatabaseQuery()`

## Method Summary

abstract DatabaseQuery	<b>and</b> (DatabaseQuery query) Create a new DatabaseQuery based upon the logical AND of the predicates represented by this query and the argument query.
abstract DatabaseQuery	<b>and</b> (DatabaseQuery query, java.lang.String contextNode) Create a new DatabaseQuery object based upon the logical AND of the predicates represented by this query and the argument query.
abstract DatabaseQuery	<b>negate</b> () Create a new DatabaseQuery, which is the logical NOT of this query.
static DatabaseQuery	<b>newInstance</b> (java.lang.String fieldName, int comparison, java.lang.String value) Make a new DatabaseQuery for a specific value in a specific field.
abstract DatabaseQuery	<b>or</b> (DatabaseQuery query) Create a new DatabaseQuery based upon the logical OR of the predicates represented by this query and the argument query.

### Methods inherited from class org.ocap.hn.content.navigation.ContentDatabaseFilter

accept

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### EQUALS

public static final int **EQUALS**

Used to specify a test for equality. For numbers and dates, an exact match is made. For strings, a case-insensitive comparison is made

**See Also:**

Constant Field Values

### GREATER\_THAN

public static final int **GREATER\_THAN**

Operator to specify greater than . For numbers the test is "a>b". For Date fields, the test is "a is after b". For strings, the comparison is based on case-insensitive dictionary ordering (i.e. which value occurs first when sorted in alphabetical order). E.g. "dog" > "cat" == true, "Chimp" > "dog" == false

**See Also:**

Constant Field Values

**LESS\_THAN**

```
public static final int LESS_THAN
```

Operator to specify less than . For numbers the test is "a<b". For Date fields, the test is "a is before b". For strings, the comparison is based on case-insensitive dictionary ordering (i.e. which one occurs first when sorted in alphabetical order).

**See Also:**

Constant Field Values

**GREATER\_THAN\_OR\_EQUALS**

```
public static final int GREATER_THAN_OR_EQUALS
```

Operator to specify greater than or equals. See rules for GREATER\_THAN and EQUALS for how non-numeric fields are compared.

**See Also:**

Constant Field Values

**LESS\_THAN\_OR\_EQUALS**

```
public static final int LESS_THAN_OR_EQUALS
```

Operator to specify less than or equals. See rules for LESS\_THAN and EQUALS for how non-numeric fields are compared.

**See Also:**

Constant Field Values

**CONTAINS**

```
public static final int CONTAINS
```

Operator to check for a String being contained within the field The comparison is case insensitive, non whole word.

**See Also:**

Constant Field Values

**NOT\_EQUALS**

```
public static final int NOT_EQUALS
```

Operator to check for in-equality. NOT\_EQUALS follows the same equality checking rules as EQUALS

**See Also:**

Constant Field Values

**EXISTS**

```
public static final int EXISTS
```

Operator to check to see if a field exists

**See Also:**

Constant Field Values

## Constructor Detail

**DatabaseQuery**

```
public DatabaseQuery()
```

## Method Detail

### newInstance

```
public static DatabaseQuery newInstance(java.lang.String fieldName,
                                         int comparison,
                                         java.lang.String value)
                                         throws DatabaseException
```

Make a new DatabaseQuery for a specific value in a specific field. This is how the root predicates of a query is formed. For example:

```
DatabaseQuery q1 = DatabaseQuery.newInstance("Title",
DatabaseQuery.CONTAINS, "Foxes");
DatabaseQuery q2 = DatabaseQuery.newInstance("Genre",
DatabaseQuery.CONTAINS, "3.4.11");
DatabaseQuery the_query = q1.and(q2);
```

#### Parameters:

**fieldName** - The name of the field to compare. This field must be the name of known field ID, i.e. FieldID.getInstance().hasKey(fieldName) must return true.

**comparison** - The type of comparison to make (CONTAINS, EXISTS, LESS\_THAN, etc)

**value** - The value to check. For numeric fields, the value must be a string that contains a number. For date fields, the value must be a date in an ISO 8601 compliant format (e.g. of the form "2001-01-05T14:30:00Z" or "2001-01-05T15:30:00+01:00"). For fields that contain an item from a classification scheme (e.g. the "Genre" field), the ID of the controlled term must be used (e.g. "3.4.11"). For comparison==DatabaseQuery.EXISTS, this parameter is not used (it is safe to pass null in this case).

#### Throws:

DatabaseException - if the specified fieldName is unknown, if the comparison is unknown or value is invalid.

### and

```
public abstract DatabaseQuery and(DatabaseQuery query)
                                   throws DatabaseException
```

Create a new DatabaseQuery based upon the logical AND of the predicates represented by this query and the argument query.

#### Parameters:

**query** - The second predicate for the AND

#### Throws:

DatabaseException - if the AND of these two queries is known to be invalid

### and

```
public abstract DatabaseQuery and(DatabaseQuery query,
                                   java.lang.String contextNode)
                                   throws DatabaseException
```

Create a new DatabaseQuery object based upon the logical AND of the predicates represented by this query and the argument query. The resulting predicate will only match within the scope of the specified node. For example:

```
DatabaseQuery a =
DatabaseQuery.newInstance("Title", DatabaseQuery.CONTAINS, "grave");
DatabaseQuery b =
DatabaseQuery.newInstance("TitleLanguage", DatabaseQuery.EQUALS, "en");
DatabaseQuery query = a.and(b, "Title");
```

will cause a search for a title that contains the name "grave" and has an English language title, within the same title node. Without the context node, a valid match would be found for:

```
Dilemme
Grave Serious
Dilemma
```

because there is a title with lang="en" and a title which contains "grave".

**Parameters:**

query - The second predicate for the AND

contextNode - The node within which all of the AND must be satisfied. This node must be the name of known field ID, (i.e. FieldID.getInstance().hasKey(fieldName) must return true) and must be a node that is at least the highest level element represented by the two predicates (i.e. the context node is not a child node of either predicate).

**Throws:**

DatabaseException - if the AND of these two queries is known to be invalid, or the contextNode is invalid

**or**

```
public abstract DatabaseQuery or(DatabaseQuery query)
    throws DatabaseException
```

Create a new DatabaseQuery based upon the logical OR of the predicates represented by this query and the argument query.

**Parameters:**

query - The second predicate for the OR

**Throws:**

DatabaseException - if the OR of these two queries is known to be invalid

**negate**

```
public abstract DatabaseQuery negate()
```

Create a new DatabaseQuery, which is the logical NOT of this query.

**Returns:**

a new DatabaseQuery object that is the logical negative of this object.

## org.ocap.hn.content.navigation Class DeviceFilter

```
java.lang.Object
├─ org.ocap.hn.content.navigation.ContentDatabaseFilter
└─ org.ocap.hn.content.navigation.DeviceFilter
```

```
public class DeviceFilter
extends ContentDatabaseFilter
```

This class represents a filtering criteria based on a particular device. Applications may use this filter to create a ContentList with content entries from a particular device.

### Constructor Summary

```
DeviceFilter(Device device)
    Creates a new DeviceFilter.
```

### Method Summary

boolean	<b>accept</b> (ContentEntry entry) Inherited from ContentDatabaseFilter.
---------	---

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### DeviceFilter

```
public DeviceFilter(Device device)
    Creates a new DeviceFilter. Only ContentItems which are hosted by the specified Device will pass this filter.
Parameters:
    device - the device to filter on
```

## Method Detail

### accept

```
public boolean accept(ContentEntry entry)
    Inherited from ContentDatabaseFilter. This function SHALL return true if the entry passed in is from the device specified in the constructor of this class.
Specified by:
    accept in class ContentDatabaseFilter
Parameters:
    entry - The entry to test for acceptance
Returns:
    true if the entry passed in is from the associated device, false otherwise.
```

## Annex D UPnP Profiles API

Package `org.ocap.hn.profiles.upnp`

### Interface Summary

**UPnPConstants**

This interface contains constants that are specific to UPnP and used in conjunction with the `org.ocap.hn.content.MetadataNode` interface.

## org.ocap.hn.profiles.upnp Interface UPnPConstants

```
public interface UPnPConstants
```

This interface contains constants that are specific to UPnP and used in conjunction with the `org.ocap.hn.content.MetadataNode` interface.

Field Summary	
<code>static java.lang.String</code>	<b>ACTOR</b> Name of an actor appearing in a video item.
<code>static java.lang.String</code>	<b>ACTOR_AT_ROLE</b> Role of an actor in the work.
<code>static java.lang.String</code>	<b>ALBUM</b> This identifies a ALBUM this piece of content belongs to.
<code>static java.lang.String</code>	<b>ALBUM_ART</b> Reference to album art.
<code>static java.lang.String</code>	<b>ARTIST</b> Name of an artist.
<code>static java.lang.String</code>	<b>ARTIST_AT_ROLE</b> Role of an artist in the work.
<code>static java.lang.String</code>	<b>ARTIST_DISCOGRAPHY</b> Reference to artist discography.
<code>static java.lang.String</code>	<b>AUTHOR</b> Name of an author.
<code>static java.lang.String</code>	<b>AUTHOR_AT_ROLE</b> Role of an author in the work (e.g. lyrics, music).
<code>static java.lang.String</code>	<b>CHANNEL_NAME</b> Used for identification of channels themselves, or information associated with a piece of recorded content.
<code>static java.lang.String</code>	<b>CHANNEL_NUMBER</b> Used for identification of tuner channels themselves or information associated with a piece of recorded content.
<code>static java.lang.String</code>	<b>COMMENTS</b> General-purpose tag where a user can annotate an object with some user-specific information.
<code>static java.lang.String</code>	<b>CONTRIBUTOR</b> Name of a contributor.
<code>static java.lang.String</code>	<b>CREATION_DATE</b> This identifies the CREATION_DATE of a piece of content.
<code>static java.lang.String</code>	<b>CREATOR</b> This identifies the CREATOR of a piece of content.

<b>Field Summary</b>	
static java.lang.String	<b>DESCRIPTION</b> A brief description of the content item.
static java.lang.String	<b>DIRECTOR</b> Name of a director.
static java.lang.String	<b>DVD_REGION_CODE</b> DVD region code.
static java.lang.String	<b>GENRE</b> Name of the genre to which an object belongs.
static java.lang.String	<b>ICON_REF</b> Reference to an icon which can be used to represent the content.
static java.lang.String	<b>ID</b> An identifier for the object.
static java.lang.String	<b>LANGUAGE</b> Language as defined by RFC 3066, e.g. "en-US".
static java.lang.String	<b>LONG_DESCRIPTION</b> A long description of the content item.
static java.lang.String	<b>LYRICS_REF</b> Reference to lyrics of a track or album.
static java.lang.String	<b>MEDIA_ID</b> Unique identifier of an audio CD (e.g. freedb or cddb id).
static java.lang.String	<b>PARENT_ID</b> An identifier for the parent of this object.
static java.lang.String	<b>PLAYLIST</b> Name of a playlist this object belongs to.
static java.lang.String	<b>PRODUCER</b> Name of a producer.
static java.lang.String	<b>PROP_STORAGE_FREE</b> Property indicating current storage space available on a storage container.
static java.lang.String	<b>PROP_STORAGE_TOTAL</b> Property indicating total storage on a storage container.
static java.lang.String	<b>PUBLISHER</b> Name of a publisher.
static java.lang.String	<b>RADIO_BAND</b> Radio station frequency band.
static java.lang.String	<b>RADIO_CALL_SIGN</b> Radio station call sign, e.g. "KSJO".
static java.lang.String	<b>RADIO_STATION_ID</b> Some identification, e.g. "107.7", broadcast frequency of the radio station.
static java.lang.String	<b>RATING</b> Rating of the object's resource, for 'parental control' filtering purposes, such as "R", "PG-13", "X".

Field Summary	
static java.lang.String	<b>REGION</b> Some identification of the region, associated with the 'source' of the object, e.g. "US", "Latin America", "Seattle".
static java.lang.String	<b>RELATION</b> Reference to related resources.
static java.lang.String	<b>RIGHTS</b> Element Description: Information about rights held in and over the resource.
static java.lang.String	<b>SCHEDULED_END_TIME</b> End time of a scheduled program.
static java.lang.String	<b>SCHEDULED_START_TIME</b> Start time of a scheduled program.
static java.lang.String	<b>STORAGE_MEDIUM</b> Indicates the type of storage medium used for the content.
static java.lang.String	<b>TITLE</b> The identifier for the title of an item.
static java.lang.String	<b>TRACK_NUMBER</b> Original track number on a CD or other medium.

## Field Detail

### ID

static final java.lang.String **ID**  
 An identifier for the object. The value of each object id property must be unique with respect to the server hosting this content.  
 The value is `didl-lite:(object)@" id"`  
**See Also:**  
 Constant Field Values

### TITLE

static final java.lang.String **TITLE**  
 The identifier for the title of an item. This could be the title of a song, a recording, a photo etc. This identifier is valid for all kinds of content.  
 Queries for **TITLE** should always return a String.  
 The value of this key is "dc:title".  
**See Also:**  
 Constant Field Values

### CREATOR

static final java.lang.String **CREATOR**  
 This identifies the **CREATOR** of a piece of content. In the case of e.g., MP3s, this maps to the 'Artist' ID3 tag, In case of a recording/live broadcast, this is the Broadcaster e.g., BBC1.  
 Queries for **CREATOR** should always return a String.

The value of this key is "dc:creator".

**See Also:**

Constant Field Values

## ARTIST

```
static final java.lang.String ARTIST
```

Name of an artist.

The value of this field is "upnp:artist".

**See Also:**

Constant Field Values

## ARTIST\_AT\_ROLE

```
static final java.lang.String ARTIST_AT_ROLE
```

Role of an artist in the work.

The value of this field is "upnp:artist@role".

**See Also:**

Constant Field Values

## ACTOR

```
static final java.lang.String ACTOR
```

Name of an actor appearing in a video item.

The value of this field is "upnp:actor".

**See Also:**

Constant Field Values

## ACTOR\_AT\_ROLE

```
static final java.lang.String ACTOR_AT_ROLE
```

Role of an actor in the work.

The value of this field is "upnp:actor@role".

getMetadata returns a String.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## AUTHOR

```
static final java.lang.String AUTHOR
```

Name of an author.

The value of this field is "upnp:author".

getMetadata() will return an array of Strings.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## AUTHOR\_AT\_ROLE

`static final java.lang.String AUTHOR_AT_ROLE`

Role of an author in the work (e.g. lyrics, music).

The value of this field is "upnp:author@role"

`getMetadata` returns a `String`.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## PRODUCER

`static final java.lang.String PRODUCER`

Name of a producer.

The value of this field is "upnp:producer".

`getMetadata()` will return an array of `Strings`.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## DIRECTOR

`static final java.lang.String DIRECTOR`

Name of a director.

The value of this field is "upnp:director".

`getMetadata()` will return an array of `Strings`.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## PUBLISHER

`static final java.lang.String PUBLISHER`

Name of a publisher.

The value of this field is "dc:publisher".

`getMetadata()` will return an array of `Strings`.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## CONTRIBUTOR

`static final java.lang.String CONTRIBUTOR`

Name of a contributor. It is recommended that `CONTRIBUTOR` includes the name of the primary content creator (see Dublin Core 'creator' property)

The value of this field is "dc:contributor".

`getMetadata()` will return an array of `Strings`.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## GENRE

`static final java.lang.String GENRE`

Name of the genre to which an object belongs. Can be more than one.

The value of this field is "upnp:genre".

`getMetadata()` will return an array of Strings.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## ALBUM

`static final java.lang.String ALBUM`

This identifies a ALBUM this piece of content belongs to. For example, in MP3 files this maps to the 'Album' ID3 tag. In case of a recording/live broadcast this could be the series to which it belongs (e.g., Buffy).

The value of this field is "upnp:album".

Queries for ALBUM should always return a String.

`getMetadata()` will return an array of Strings.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## PLAYLIST

`static final java.lang.String PLAYLIST`

Name of a playlist this object belongs to. Can be more than one.

The value of this field is "upnp:playlist".

`getMetadata()` will return an array of Strings.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## ALBUM\_ART

`static final java.lang.String ALBUM_ART`

Reference to album art. Can be more than one.

The value of this field is "upnp:albumArtURI".

Values must be properly escaped URIs as described in [RFC 2396].

`getMetadata()` will return an array of Strings.

**See Also:**

`MetadataNode.getMetadata(String)`, Constant Field Values

## ARTIST\_DISCOGRAPHY

`static final java.lang.String ARTIST_DISCOGRAPHY`

Reference to artist discography.

The value of this field is "upnp:artistDiscographyURI".

Values must be properly escaped URIs as described in [RFC 2396].

`getMetadata()` will return a String.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

### LYRICS\_REF

static final java.lang.String **LYRICS\_REF**

Reference to lyrics of a track or album.

The value of this field is "upnp:lyricsURI".

Values must be properly escaped URIs as described in [RFC 2396].

getMetadata() will return an array of Strings.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

### RELATION

static final java.lang.String **RELATION**

Reference to related resources.

The value of this field is "dc:relation".

Values must be properly escaped URIs as described in [RFC 2396].

getMetadata() will return an array of Strings.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

### STORAGE\_MEDIUM

static final java.lang.String **STORAGE\_MEDIUM**

Indicates the type of storage medium used for the content. Potentially useful for user-interface purposes. Allowed values are defined by UPnP and include:

- "UNKNOWN"
- "DV"
- "MINI-DV"
- "VHS"
- "W-VHS"
- "S-VHS"
- "D-VHS"
- "VHSC"
- "VIDEO8"
- "HI8"
- "CD-ROM"
- "CD-DA"
- "CD-R"
- "CD-RW"

- "VIDEO-CD"
- "SACD"
- "MD-AUDIO"
- "MD-PICTURE"
- "DVD-ROM"
- "DVD-VIDEO"
- "DVD-R"
- "DVD+RW"
- "DVD-RW"
- "DVD-RAM"
- "DVD-AUDIO"
- "DAT"
- "LD"
- "HDD"
- "SD"
- "PC-CARD"
- "MMC"
- "CF"
- "BD"
- "MS"

The value of this field is "upnp:storageMedium".

**See Also:**

Constant Field Values

## DESCRIPTION

static final java.lang.String **DESCRIPTION**

A brief description of the content item.

The value of this field is "dc:description".

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## LONG\_DESCRIPTION

static final java.lang.String **LONG\_DESCRIPTION**

A long description of the content item.

The value of this field is "upnp:longDescription".

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## ICON\_REF

static final java.lang.String **ICON\_REF**

Reference to an icon which can be used to represent the content.

The value of this field is "upnp:icon".

Values must be properly escaped URIs as described in [RFC 2396].

IB: Do we want a key/value pair icon -> java.awt.Image ?

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## REGION

static final java.lang.String **REGION**

Some identification of the region, associated with the 'source' of the object, e.g. "US", "Latin America", "Seattle".

The value of this field is "upnp:region"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## RATING

static final java.lang.String **RATING**

Rating of the object's resource, for 'parental control' filtering purposes, such as "R", "PG-13", "X".

The value of this field is "upnp:rating"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## RIGHTS

static final java.lang.String **RIGHTS**

Element Description: Information about rights held in and over the resource. Typically a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource. Guidelines for content creation: The Rights element may be used for either a textual statement or a URL pointing to a rights statement, or a combination, when a brief statement and a more lengthy one are available. Examples:

Rights="Access limited to members"

Rights="http://cs-tr.cs.cornell.edu/Dienst/Repository/2.0/Terms"

The value of this field is "dc:rights"

getMetadata() returns an array of Strings.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## CREATION\_DATE

static final java.lang.String **CREATION\_DATE**

This identifies the CREATION\_DATE of a piece of content. In the case of e.g., MP3's this maps to the 'Year' ID3 tag, In case of a recording/live broadcast this is when the content was created. For Images this is the date the photo was made.

Queries for CREATION\_DATE should always return a java.util.Date. Only the year of the Date might actually be valid (e.g., for MP3s).

The value of this field is "dc:date"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## LANGUAGE

static final java.lang.String **LANGUAGE**

Language as defined by RFC 3066, e.g. "en-US".

The value of this field is "dc:language"

getMetadata() will return an array of Strings.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## RADIO\_CALL\_SIGN

static final java.lang.String **RADIO\_CALL\_SIGN**

Radio station call sign, e.g. "KSJO".

The value of this field is "upnp:radioCallSign"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## RADIO\_STATION\_ID

static final java.lang.String **RADIO\_STATION\_ID**

Some identification, e.g. "107.7", broadcast frequency of the radio station.

The value of this field is "upnp:radioStationID"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## RADIO\_BAND

static final java.lang.String **RADIO\_BAND**

Radio station frequency band. Recommended values are "AM", "FM", "Shortwave", "Internet", "Satellite". Vendor's may extend this.

The value of this field is "upnp:radioBand"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## CHANNEL\_NUMBER

static final java.lang.String **CHANNEL\_NUMBER**

Used for identification of tuner channels themselves or information associated with a piece of recorded content.

The value of this field is "upnp:channelNr"

**getMetadata() returns an Integer.**

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## CHANNEL\_NAME

**static final java.lang.String CHANNEL\_NAME**

Used for identification of channels themselves, or information associated with a piece of recorded content.

The value of this field is "upnp:channelName"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## SCHEDULED\_START\_TIME

**static final java.lang.String SCHEDULED\_START\_TIME**

Start time of a scheduled program.

The value of this field is "upnp:scheduledStartTime"

getMetadata() returns java.util.Date.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## SCHEDULED\_END\_TIME

**static final java.lang.String SCHEDULED\_END\_TIME**

End time of a scheduled program.

The value of this field is "upnp:scheduledEndTime"

getMetadata() returns java.util.Date.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## DVD\_REGION\_CODE

**static final java.lang.String DVD\_REGION\_CODE**

DVD region code.

The value of this field is "upnp:DVDRegionCode"

getMetadata() returns an Integer.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## TRACK\_NUMBER

**static final java.lang.String TRACK\_NUMBER**

Original track number on a CD or other medium.

The value of this field is "upnp:originalTrackNumber"

getMetadata() returns an Integer.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## MEDIA\_ID

**static final java.lang.String MEDIA\_ID**

Unique identifier of an audio CD (e.g. freedb or cddb id).

The value of this field is "upnp:toc"

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## COMMENTS

static final java.lang.String **COMMENTS**

General-purpose tag where a user can annotate an object with some user-specific information.

The value of this field is "upnp:userAnnotation"

getMetadata() will return an array of Strings.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## PROP\_STORAGE\_TOTAL

static final java.lang.String **PROP\_STORAGE\_TOTAL**

Property indicating total storage on a storage container.

The value of this field is "upnp:storageTotal"

getMetadata() will return a Long.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## PROP\_STORAGE\_FREE

static final java.lang.String **PROP\_STORAGE\_FREE**

Property indicating current storage space available on a storage container.

The value of this field is "upnp:storageFree"

getMetadata() will return a Long.

**See Also:**

MetadataNode.getMetadata(String), Constant Field Values

## PARENT\_ID

static final java.lang.String **PARENT\_ID**

An identifier for the parent of this object.

The value is didl-lite:(object)@"parentID"

**See Also:**

Constant Field Values

## Annex E Service API

Package `org.ocap.hn.service`

### Interface Summary

<b>RemoteService</b>	A RemoteService is a service which is hosted or provided by another device on the home network.
----------------------	---

## org.ocap.hn.service Interface RemoteService

### All Superinterfaces:

javax.tv.service.Service

```
public interface RemoteService
extends javax.tv.service.Service
```

A RemoteService is a service which is hosted or provided by another device on the home network.

## Method Summary

ContentItem	<b>getContentItem()</b> Returns the ContentItem associated with this remote service
-------------	--

### Methods inherited from interface javax.tv.service.Service

equals, getLocator, getName, getServiceType, hashCode, hasMultipleInstances, retrieveDetails

## Method Detail

### getContentItem

```
ContentItem getContentItem()
```

Returns the ContentItem associated with this remote service

**Returns:**  
The ContentItem associated with this service.

## Annex F Recording API

### Package org.ocap.hn.recording

#### Interface Summary

<b>NetRecordingEntry</b>	This ContentEntry represents a series recording that has been scheduled on the home network.
<b>NetRecordingRequestHandler</b>	A class implementing this interface processes recording requests received from devices on the home network.
<b>NetRecordingRequestManager</b>	This interface represents a local RecordingNetModule.
<b>RecordingContentItem</b>	This ContentItem represents a recording that has been scheduled on the home network.
<b>RecordingNetModule</b>	An interface representing a NetModule which provides DVR functionality.

#### Class Summary

<b>NetRecordingSpec</b>	This class represents a network recording specification.
-------------------------	--

## org.ocap.hn.recording Interface NetRecordingEntry

**All Superinterfaces:**  
ContentEntry

```
public interface NetRecordingEntry
extends ContentEntry
```

This ContentEntry represents a series recording that has been scheduled on the home network.

### Field Summary

static java.lang.String	<b>PROP_NET_RECORDING_ENTRY</b> Key for returning the property which indicates that this entry is a NetRecordingEntry.
-------------------------	---

### Method Summary

void	<b>addRecordingContentItem</b> (RecordingContentItem item) Adds a local RecordingContentItem to this recording object
java.lang.String[]	<b>getRecordingContentItemIDs</b> () Retrieves ObjectIDs of the individual RecordingContentItems that make up this series recording.
RecordingContentItem[]	<b>getRecordingContentItems</b> () Retrieves the local individual RecordingContentItems that make up this series recording.
void	<b>removeRecordingContentItem</b> (RecordingContentItem item) Removes a local RecordingContentItem from this recording object.

### Methods inherited from interface org.ocap.hn.content.ContentEntry

deleteEntry, getContentSize, getCreationDate, getEntryParent, getExtendedFileAccessPermissions, getID, getParentID, getRootMetadataNode, getServer, isLocal

### Field Detail

#### PROP\_NET\_RECORDING\_ENTRY

```
static final java.lang.String PROP_NET_RECORDING_ENTRY
```

Key for returning the property which indicates that this entry is a NetRecordingEntry. Values returned for this key will be represented as a Boolean.

**See Also:**

Constant Field Values

## Method Detail

### getRecordingContentItems

RecordingContentItem[] **getRecordingContentItems**()  
throws java.io.IOException

Retrieves the local individual RecordingContentItems that make up this series recording.

**Returns:**

the RecordingContentItems in this series

**Throws:**

java.io.IOException - if this isLocal() method of this object does not return true

### addRecordingContentItem

void **addRecordingContentItem**(RecordingContentItem item)  
throws java.io.IOException

Adds a local RecordingContentItem to this recording object

**Parameters:**

item - The recording content item to add to this series

**Throws:**

java.io.IOException - if this isLocal() method of this object does not return true

java.lang.IllegalStateException - if this recording object is not associated with a UPnP AV Scheduled Recording Service Object (RecordSchedule)

java.lang.IllegalArgumentException - if the RecordingContentItem parameter has the associated UPnP AV Scheduled Recording Service Object (RecordTask)

java.lang.SecurityException - if the caller does not have HomeNetPermission("recordinghandler")

### removeRecordingContentItem

void **removeRecordingContentItem**(RecordingContentItem item)  
throws java.io.IOException

Removes a local RecordingContentItem from this recording object. If the RecordingContentItem passed into this method is not contained in this NetRecordingObject, this method has no effect.

**Parameters:**

item - The recording content item to remove from this series

**Throws:**

java.io.IOException - if this isLocal() method of this object does not return true

java.lang.IllegalArgumentException - if the RecordingContentItem parameter has the associated UPnP AV Scheduled Recording Service Object (RecordTask)

java.lang.SecurityException - if the caller does not have HomeNetPermission("recordinghandler")

### getRecordingContentItemIDs

java.lang.String[] **getRecordingContentItemIDs**()

Retrieves ObjectIDs of the individual RecordingContentItems that make up this series recording.

**Returns:**

the ObjectIDs of the RecordingContentItems in this series

## org.ocap.hn.recording Interface NetRecordingRequestHandler

public interface **NetRecordingRequestHandler**

A class implementing this interface processes recording requests received from devices on the home network.

An application which has a class implementing this interface may set an instance of it in a local RecordingNetModule. It is up to the application to interpret metadata associated with NetRecordingSpecs and RecordingContentItems delivered in the callback methods, and translate these requests into local DVR recordings.

### Method Summary

boolean	<b>notifyDelete</b> (java.net.InetAddress address, ContentEntry recording) Notifies this NetRecordingRequestHandler that a device on the home network has requested that metadata associated with a recording be deleted.
boolean	<b>notifyDeleteService</b> (java.net.InetAddress address, ContentEntry recording) Notifies this NetRecordingRequestHandler that a device on the home network has requested that content associated with a recorded service be deleted.
boolean	<b>notifyDisable</b> (java.net.InetAddress address, ContentEntry recording) Notifies this NetRecordingRequestHandler that a device on the home network has requested that a recording be disabled.
boolean	<b>notifyPrioritization</b> (java.net.InetAddress address, NetRecordingEntry[] recordings) Notifies this NetRecordingRequestHandler that a device on the home network has requested that a group of recordings be re-prioritized.
boolean	<b>notifyPrioritization</b> (java.net.InetAddress address, RecordingContentItem[] recordings) Notifies this NetRecordingRequestHandler that a device on the home network has requested that a group of individual recordings be re-prioritized.
boolean	<b>notifyReschedule</b> (java.net.InetAddress address, ContentEntry recording, NetRecordingEntry spec) Notifies this NetRecordingRequestHandler that a device on the home network has requested that a recording be rescheduled.
boolean	<b>notifySchedule</b> (java.net.InetAddress address, NetRecordingEntry spec) Notifies this NetRecordingRequestHandler that a device on the home network has requested that a recording be scheduled.

### Method Detail

#### notifySchedule

boolean **notifySchedule**(java.net.InetAddress address, NetRecordingEntry spec)

Notifies this NetRecordingRequestHandler that a device on the home network has requested that a recording be scheduled. Handler applications MAY inspect any metadata associated with the NetRecordingEntry passed with the method invocation, and translate such metadata in one or more local

DVR recordings. Applications SHOULD associate such recordings with the `NetRecordingEntry` by adding the recordings to the entry using the `NetRecordingEntry.addRecordingContentItem()` method.

**Parameters:**

`address` - IP address of the device on the home network which has issues this request  
`spec` - the `NetRecordingEntry` which describes the requested recording

**Returns:**

true if the schedule request can be successfully processed, or false if the request will not be processed.

**See Also:**

`NetRecordingEntry.addRecordingContentItem(RecordingContentItem)`

### notifyReschedule

```
boolean notifyReschedule(java.net.InetAddress address,
                        ContentEntry recording,
                        NetRecordingEntry spec)
```

Notifies this `NetRecordingRequestHandler` that a device on the home network has requested that a recording be rescheduled. Handler applications MAY inspect any metadata contained within the `NetRecordingEntry` passed into this method, and utilize such metadata to reschedule the local DVR recording represented by the given `ContentEntry`. This `ContentEntry` may represent an individual recording as a `RecordingContentItem`, or may represent a collection of recordings contained within a `NetRecordingEntry` object.

**Parameters:**

`address` - the IP address of the device on the home network which has issues this request  
`recording` - the `RecordingContentItem` or `NetRecordingEntry` to be rescheduled  
`spec` - the `NetRecordingEntry` object containing the metadata to be used to reschedule the recording

**Returns:**

true if the reschedule request can be successfully processed, or false if the request will not be processed.

### notifyDisable

```
boolean notifyDisable(java.net.InetAddress address,
                     ContentEntry recording)
```

Notifies this `NetRecordingRequestHandler` that a device on the home network has requested that a recording be disabled. If the recording is in progress, this is a request to stop the recording. If the recording is pending, this is a request to cancel the recording. Applications MAY cancel or stop the given recording in response to this request.

**Parameters:**

`address` - the IP address of the device on the home network which has issues this request  
`recording` - the `RecordingContentItem` or `RecordingNetEntry` to be disabled

**Returns:**

true if the disable request can be successfully processed, or false if the request will not be processed.

### notifyDelete

```
boolean notifyDelete(java.net.InetAddress address,
                    ContentEntry recording)
```

Notifies this `NetRecordingRequestHandler` that a device on the home network has requested that metadata associated with a recording be deleted. Applications MAY delete the given recording in response to this request.

**Parameters:**

`address` - the IP address of the device on the home network which has issues this request  
`recording` - the `RecordingContentItem` or `NetRecordingEntry` to be deleted

**Returns:**

true if the delete request can be successfully processed, or false if the request will not be processed.

**notifyDeleteService**

```
boolean notifyDeleteService(java.net.InetAddress address,  
                             ContentEntry recording)
```

Notifies this NetRecordingRequestHandler that a device on the home network has requested that content associated with a recorded service be deleted. Applications MAY delete the content associated with the given recording in response to this request.

**Parameters:**

address - the IP address of the device on the home network which has issues this request  
recording - requested the RecordingContentItem or NetRecordingEntry

**Returns:**

true if the delete request can be successfully processed, or false if the request will not be processed.

**notifyPrioritization**

```
boolean notifyPrioritization(java.net.InetAddress address,  
                              NetRecordingEntry[] recordings)
```

Notifies this NetRecordingRequestHandler that a device on the home network has requested that a group of recordings be re-prioritized. The requested prioritization is represented by the ordering of the NetRecordingEntry objects in the given array, with the highest priority at index 0 of the array. Applications MAY prioritize some or all of the local DVR recordings contained within the NetRecordingEntry array.

**Parameters:**

address - the IP address of the device on the home network which has issues this request  
recordings - the NetRecordingEntries to be prioritized

**Returns:**

true if the prioritization request can be successfully processed, or false if the request will not be processed.

**notifyPrioritization**

```
boolean notifyPrioritization(java.net.InetAddress address,  
                              RecordingContentItem[] recordings)
```

Notifies this NetRecordingRequestHandler that a device on the home network has requested that a group of individual recordings be re-prioritized. The requested prioritization is represented by the ordering of the RecordingContentItem objects in the given array, with the highest priority at index 0 of the array. Applications MAY prioritize the local DVR recordings contained within the RecordingContentItem array.

**Parameters:**

address - IP address of the device on the home network which has issued this request.  
recordings - The recording content items associated with the recordings to be prioritized.

**Returns:**

True if the prioritization request can be successfully processed, or false if the request will not be processed.

## org.ocap.hn.recording Interface NetRecordingRequestManager

### All Superinterfaces:

NetModule, RecordingNetModule

```
public interface NetRecordingRequestManager
extends RecordingNetModule
```

This interface represents a local RecordingNetModule. An instance of RecordingNetModule for which the isLocal() method returns true will also be an instance of NetRecordingRequestManager.

## Field Summary

### Fields inherited from interface org.ocap.hn.NetModule

CONTENT\_LIST, CONTENT\_MANAGER, CONTENT\_RECORDER, CONTENT\_RENDERER,  
CONTENT\_SERVER, PROP\_CONTROL\_URL, PROP\_DESCRIPTION\_URL, PROP\_EventSub\_URL,  
PROP\_NETMODULE\_ID, PROP\_NETMODULE\_TYPE

## Method Summary

NetRecordingEntry	<b>createNetRecordingEntry()</b> This method creates a local entry which represents a network visible collection of recording items.
void	<b>setNetRecordingRequestHandler</b> (NetRecordingRequestHandler handler) This method sets the specified NetRecordingRequestHandler that processes requests to schedule recordings from remote devices.

### Methods inherited from interface org.ocap.hn.recording.RecordingNetModule

requestDelete, requestDeleteService, requestDisable, requestPrioritize,  
requestPrioritize, requestReschedule, requestSchedule

### Methods inherited from interface org.ocap.hn.NetModule

addNetModuleEventListener, getDevice, getKeys, getNetModuleId,  
getNetModuleType, getProperty, isLocal, removeNetModuleEventListener

## Method Detail

### createNetRecordingEntry

```
NetRecordingEntry createNetRecordingEntry()
```

throws java.io.IOException

This method creates a local entry which represents a network visible collection of recording items.

#### Throws:

java.io.IOException - if the isLocal() method of this object does not return true

java.lang.SecurityException - if the caller does not have HomeNetPermission("recordinghandler")

### **setNetRecordingRequestHandler**

void **setNetRecordingRequestHandler**(NetRecordingRequestHandler handler)

This method sets the specified NetRecordingRequestHandler that processes requests to schedule recordings from remote devices. Only one instance of NetRecordingRequestHandler can be set on a given RecordingNetModule at a time. A NetRecordingRequestHandler can only be set on an instance of RecordingNetModule that is local to the device.

**Parameters:**

handler - the handler to be set for this RecordingNetModule. If null is specified, the currently set handler will be removed, and no application notification will occur for recording requests.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("recordinghandler")

## org.ocap.hn.recording Class NetRecordingSpec

```
java.lang.Object
└─ org.ocap.hn.recording.NetRecordingSpec
```

```
public class NetRecordingSpec
extends java.lang.Object
```

This class represents a network recording specification. NetRecordingSpec object may be used to request recordings be scheduled on remote devices. Metadata contained within this object can be used to schedule or modify recordings on the home network.

### Constructor Summary

**NetRecordingSpec()**

Default constructor for the NetRecordingSpec.

**NetRecordingSpec**(MetadataNode metadata)

Metadata constructor for the NetRecordingSpec.

### Method Summary

MetadataNode	<b>getMetadata()</b>
--------------	----------------------

Retrieves the root metadata node for this recording spec.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### NetRecordingSpec

```
public NetRecordingSpec()
```

Default constructor for the NetRecordingSpec. Upon creation, NetRecordingSpecs will contain a single empty metadata node.

#### NetRecordingSpec

```
public NetRecordingSpec(MetadataNode metadata)
```

Metadata constructor for the NetRecordingSpec. Applications can use this form of the constructor to specify the root metadata node at time of construction.

**Parameters:**

metadata - root metadata node for the NetRecordingSpec

## Method Detail

### **getMetadata**

```
public MetadataNode getMetadata()
```

Retrieves the root metadata node for this recording spec. Metadata added to this recording spec will be utilized to identify recording requests on remote devices.

**Returns:**

The root MetadataNode for this NetRecordingSpec

## org.ocap.hn.recording Interface RecordingContentItem

### All Superinterfaces:

ContentEntry, ContentItem

```
public interface RecordingContentItem
extends ContentItem
```

This ContentItem represents a recording that has been scheduled on the home network. This interface represents a DVR recording which can be published to the home network. On devices which support both the OCAP Home Networking API and the OCAP DVR API, objects implementing `org.ocap.dvr.OcapRecordingRequest` will also implement this interface. When a RecordingRequest is deleted, implementations SHALL call the `RecordingContentItem.deleteEntry` method in the same object.

Field Summary	
static java.lang.String	<b>PROP_ACCESS_PERMISSIONS</b> Key constant for retrieving the file access permissions of this recording item from this item's metadata.
static java.lang.String	<b>PROP_APP_ID</b> Key constant for retrieving the application ID of this recording item from this item's metadata.
static java.lang.String	<b>PROP_CONTENT_URI</b> Key constant for retrieving the location of content associated with this recording item from this item's metadata.
static java.lang.String	<b>PROP_DELETION_REASON</b> Key constant for retrieving the reason this recording had been deleted.
static java.lang.String	<b>PROP_DELETION_TIME</b> Key constant for retrieving the time at which content associated with this recording was deleted.
static java.lang.String	<b>PROP_DESTINATION</b> Key constant for retrieving the destination of this recording item from this item's metadata.
static java.lang.String	<b>PROP_DURATION</b> Key constant for retrieving the duration in milliseconds of this recording item from this item's metadata.
static java.lang.String	<b>PROP_EXPIRATION_PERIOD</b> Key constant for retrieving the expiration period for this recording item from this item's metadata.
static java.lang.String	<b>PROP_FAILURE_REASON</b> Key constant for retrieving the reason this recording has failed.
static java.lang.String	<b>PROP_MSO_CONTENT</b> Key constant for retrieving the MSO content indicator for this recording item from this item's metadata.

<b>Field Summary</b>	
static java.lang.String	<b>PROP_NET_RECORDING_ENTRY</b> Key constant for retrieving the ID of any NetRecordingEntry containing this RecordingContentItem.
static java.lang.String	<b>PROP_ORGANIZATION</b> Key constant for retrieving the organization of this recording item from this item's metadata.
static java.lang.String	<b>PROP_PRESENTATION_POINT</b> Key constant for retrieving the presentation point for this recording item from this item's metadata.
static java.lang.String	<b>PROP_PRIORITY_FLAG</b> Key constant for retrieving the priority flag of this recording item from this item's metadata.
static java.lang.String	<b>PROP_RECORDING_STATE</b> Key constant for retrieving the state of this recording item from this item's metadata.
static java.lang.String	<b>PROP_RETENTION_PRIORITY</b> Key constant for retrieving the retention priority of this recording item from this item's metadata.
static java.lang.String	<b>PROP_SOURCE_ID</b> Key constant for retrieving the source ID of this recording item from this item's metadata.
static java.lang.String	<b>PROP_SPACE_REQUIRED</b> Key constant for retrieving the estimated space required for this recording item from this item's metadata.
static java.lang.String	<b>PROP_START_TIME</b> Key constant for retrieving the start time of this recording item from this item's metadata.

#### Fields inherited from interface org.ocap.hn.content.ContentItem

AUDIO\_ITEM, AUDIO\_ITEM\_BOOK, AUDIO\_ITEM\_BROADCAST, AUDIO\_ITEM\_TRACK, IMAGE\_ITEM, IMAGE\_ITEM\_PHOTO, ITEM, VIDEO\_ITEM, VIDEO\_ITEM\_BROADCAST, VIDEO\_ITEM\_MOVIE, VIDEO\_ITEM\_MUSIC\_CLIP

## Method Summary

NetRecordingEntry	<b>getRecordingEntry()</b> Returns the NetRecordingEntry which contains this recording content item if the NetRecordingEntry is available.
java.lang.String	<b>getRecordingEntryID()</b> Returns the ObjectID of the NetRecordingEntry which contains this recording content item.
NetActionRequest	<b>requestConflictingRecordings(NetActionHandler handler)</b> Requests a list of recordings whose usage of resources conflict with this recording content item.

## Method Summary

NetActionRequest	<b>requestSetMediaTime</b> (javax.media.Time time, NetActionHandler handler) Requests that the presentation point of this recording be updated.
------------------	---

### Methods inherited from interface org.ocap.hn.content.ContentItem

containsResource, deleteEntry, getContentClass, getItemService,  
 getRenderableResources, getResource, getResourceCount, getResourceIndex,  
 getResources, getTitle, hasAudio, hasStillImage, hasVideo, isRenderable

### Methods inherited from interface org.ocap.hn.content.ContentEntry

getContentSize, getCreationDate, getEntryParent,  
 getExtendedFileAccessPermissions, getID, getParentID, getRootMetadataNode,  
 getServer, isLocal

## Field Detail

### PROP\_RECORDING\_STATE

static final java.lang.String **PROP\_RECORDING\_STATE**

Key constant for retrieving the state of this recording item from this item's metadata. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

### PROP\_START\_TIME

static final java.lang.String **PROP\_START\_TIME**

Key constant for retrieving the start time of this recording item from this item's metadata. Values returned for this key will be represented as a java.util.Date.

**See Also:**

Constant Field Values

### PROP\_DURATION

static final java.lang.String **PROP\_DURATION**

Key constant for retrieving the duration in milliseconds of this recording item from this item's metadata. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

### PROP\_SOURCE\_ID

static final java.lang.String **PROP\_SOURCE\_ID**

Key constant for retrieving the source ID of this recording item from this item's metadata. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

**PROP\_DESTINATION**

```
static final java.lang.String PROP_DESTINATION
```

Key constant for retrieving the destination of this recording item from this item's metadata. Values returned for this key will be represented as an org.ocap.hn.content.ContentContainer.

**See Also:**

Constant Field Values

**PROP\_PRIORITY\_FLAG**

```
static final java.lang.String PROP_PRIORITY_FLAG
```

Key constant for retrieving the priority flag of this recording item from this item's metadata. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

**PROP\_RETENTION\_PRIORITY**

```
static final java.lang.String PROP_RETENTION_PRIORITY
```

Key constant for retrieving the retention priority of this recording item from this item's metadata. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

**PROP\_ACCESS\_PERMISSIONS**

```
static final java.lang.String PROP_ACCESS_PERMISSIONS
```

Key constant for retrieving the file access permissions of this recording item from this item's metadata. Values returned for this key will be represented as an org.ocap.storage.ExtendedFileAccessPermissions.

**See Also:**

Constant Field Values

**PROP\_ORGANIZATION**

```
static final java.lang.String PROP_ORGANIZATION
```

Key constant for retrieving the organization of this recording item from this item's metadata. Values returned for this key will be represented as a String.

**See Also:**

Constant Field Values

**PROP\_APP\_ID**

```
static final java.lang.String PROP_APP_ID
```

Key constant for retrieving the application ID of this recording item from this item's metadata. Values returned for this key will be represented as an org.dvb.application.AppID.

**See Also:**

Constant Field Values

**PROP\_SPACE\_REQUIRED**

```
static final java.lang.String PROP_SPACE_REQUIRED
```

Key constant for retrieving the estimated space required for this recording item from this item's metadata. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

**PROP\_CONTENT\_URI**

```
static final java.lang.String PROP_CONTENT_URI
```

Key constant for retrieving the location of content associated with this recording item from this item's metadata. Values returned for this key will be represented as an array of Strings.

**See Also:**

Constant Field Values

**PROP\_PRESENTATION\_POINT**

```
static final java.lang.String PROP_PRESENTATION_POINT
```

Key constant for retrieving the presentation point for this recording item from this item's metadata. Values returned for this key will be represented as a Long.

**See Also:**

Constant Field Values

**PROP\_EXPIRATION\_PERIOD**

```
static final java.lang.String PROP_EXPIRATION_PERIOD
```

Key constant for retrieving the expiration period for this recording item from this item's metadata. Values returned for this key will be represented as an Long.

**See Also:**

Constant Field Values

**PROP\_MSO\_CONTENT**

```
static final java.lang.String PROP_MSO_CONTENT
```

Key constant for retrieving the MSO content indicator for this recording item from this item's metadata. Values returned for this key will be represented as an Boolean.

**See Also:**

Constant Field Values

**PROP\_DELETION\_REASON**

```
static final java.lang.String PROP_DELETION_REASON
```

Key constant for retrieving the reason this recording had been deleted. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

**PROP\_DELETION\_TIME**

```
static final java.lang.String PROP_DELETION_TIME
```

Key constant for retrieving the time at which content associated with this recording was deleted. Values returned for this key will be represented as a java.util.Date.

**See Also:**

Constant Field Values

**PROP\_FAILURE\_REASON**

```
static final java.lang.String PROP_FAILURE_REASON
```

Key constant for retrieving the reason this recording has failed. Values returned for this key will be represented as an Integer.

**See Also:**

Constant Field Values

## PROP\_NET\_RECORDING\_ENTRY

static final java.lang.String **PROP\_NET\_RECORDING\_ENTRY**

Key constant for retrieving the ID of any NetRecordingEntry containing this RecordingContentItem. Values returned for this key will be represented as a String.

**See Also:**

Constant Field Values

## Method Detail

### requestSetMediaTime

NetActionRequest **requestSetMediaTime**(javax.media.Time time, NetActionHandler handler)

Requests that the presentation point of this recording be updated.

**Parameters:**

time - The presentation point of this recording.

handler - The NetActionHandler which gets informed once this request completes.

**Returns:**

NetActionRequest which can be used to monitor asynchronous action progress

### requestConflictingRecordings

NetActionRequest **requestConflictingRecordings**(NetActionHandler handler)

Requests a list of recordings whose usage of resources conflict with this recording content item. The resulting list of recordings SHALL be returned as an array of RecordingContentItem objects from the NetActionEvent.getResponse() method of the resulting NetActionEvent.

**Parameters:**

handler - The NetActionHandler implementation to receive the asynchronous response to this request

**Returns:**

NetActionRequest which can be used to monitor asynchronous action progress

### getRecordingEntry

NetRecordingEntry **getRecordingEntry**()

Returns the NetRecordingEntry which contains this recording content item if the NetRecordingEntry is available.

**Returns:**

null if this RecordingContentItem is not added to any NetRecordingEntry or if the NetRecordingEntry containing this RecordingContentItem is not available. Otherwise the NetRecordingEntry containing this RecordingContentItem

### getRecordingEntryID

java.lang.String **getRecordingEntryID**()

Returns the ObjectID of the NetRecordingEntry which contains this recording content item. The ObjectID can be obtained from ocap:netRecordingEntry property of this recording content item.

**Returns:**

null if this RecordingContentItem does not contain ocap:netRecordingEntry property. Otherwise, the value contained in ocap:netRecordingEntry property of this RecordingContentItem.

**org.ocap.hn.recording**  
**Interface RecordingNetModule**

**All Superinterfaces:**

NetModule

**All Known Subinterfaces:**

NetRecordingRequestManager

```
public interface RecordingNetModule
extends NetModule
```

An interface representing a NetModule which provides DVR functionality.

NetModules which implement this interface SHALL have a NetModule.PROP\_NETMODULE\_TYPE property value of NetModule.CONTENT\_RECORDER.

**Field Summary**

Fields inherited from interface org.ocap.hn.NetModule	
CONTENT_LIST, CONTENT_MANAGER, CONTENT_RECORDER, CONTENT_RENDERER, CONTENT_SERVER, PROP_CONTROL_URL, PROP_DESCRIPTION_URL, PROP_EventSub_URL, PROP_NETMODULE_ID, PROP_NETMODULE_TYPE	

**Method Summary**

NetActionRequest	<b>requestDelete</b> (ContentEntry recording, NetActionHandler handler) Requests that metadata associated with a scheduled recording be deleted from storage.
NetActionRequest	<b>requestDeleteService</b> (ContentEntry recording, NetActionHandler handler) Requests that content associated with a scheduled recording be deleted from storage.
NetActionRequest	<b>requestDisable</b> (ContentEntry recording, NetActionHandler handler) Requests that an in progress recording be disabled on this network recording device.
NetActionRequest	<b>requestPrioritize</b> (NetRecordingEntry[] recordings, NetActionHandler handler) Requests that a group of scheduled recording request be prioritized on this network recording device, where each recording request may represent one or more individual recordings on the remote device.
NetActionRequest	<b>requestPrioritize</b> (RecordingContentItem[] recordings, NetActionHandler handler) Requests that a group of scheduled individual recordings be prioritized on this network recording device.

## Method Summary

NetActionRequest	<b>requestReschedule</b> (ContentEntry recording, NetRecordingSpec recordingSpec, NetActionHandler handler) Requests that a recording be rescheduled on this network recording device.
NetActionRequest	<b>requestSchedule</b> (NetRecordingSpec recordingSpec, NetActionHandler handler) Requests that a recording be scheduled on this network recording device.

### Methods inherited from interface org.ocap.hn.NetModule

addNetModuleEventListener, getDevice, getKeys, getNetModuleId, getNetModuleType, getProperty, isLocal, removeNetModuleEventListener

## Method Detail

### requestSchedule

NetActionRequest **requestSchedule**(NetRecordingSpec recordingSpec, NetActionHandler handler)

Requests that a recording be scheduled on this network recording device. metadata added to the NetRecordingSpec prior to calling this method will be utilized by the remote device in identifying the recording or recordings to be scheduled. Upon completion of this operation, a NetActionEvent SHALL be delivered to the given handler indicating success or failure. Upon success, values returned by calls to NetActionEvent.getResponse() SHALL contain a NetRecordingEntry representing the newly created recording.

**Parameters:**

recordingSpec - a recording spec containing the metadata used to identify the recordings to be scheduled.

handler - The NetActionHandler which gets informed once this request completes.

**Returns:**

NetActionRequest to inform calling application of results.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("recording")

java.lang.IllegalArgumentException - if recordingSpec has an empty MetadataNode, or if MetadataNode which is associated with recordingSpec does not contain the necessary metadata entry such as scheduledChannelID, scheduledStartDateTime, scheduledDuration

### requestReschedule

NetActionRequest **requestReschedule**(ContentEntry recording, NetRecordingSpec recordingSpec, NetActionHandler handler)

Requests that a recording be rescheduled on this network recording device. Metadata added to the NetRecordingSpec prior to calling this method will be utilized by the remote device in identifying changes the recording or recordings to be rescheduled. Upon completion of this operation, a NetActionEvent SHALL be delivered to the given handler indicating success or failure.

**Parameters:**

recording - the previously scheduled RecordingContentItem or NetRecordingEntry to be rescheduled.

recordingSpec - a recording spec containing the metadata used to identify the changes to recordings to be rescheduled.

handler - The NetActionHandler which gets informed once this request completes.

**Returns:**

NetActionRequest to inform calling application of results.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("recording")  
 java.lang.IllegalArgumentException - if the recording parameter is neither the  
 NetRecordingEntry with upnp:srsRecordScheduleID metadata entry nor the RecordingContentItem with  
 upnp:srsRecordTaskID metadata entry in its own MetadataNode, or if recordingSpec has an empty  
 MetadataNode, or if MetadataNode which is associated with recordingSpec does not contain the necessary  
 metadata entry such as scheduledChannelID, scheduledStartTime, scheduledDuration

### requestDisable

NetActionRequest **requestDisable**(ContentEntry recording,  
 NetActionHandler handler)

Requests that an in progress recording be disabled on this network recording device. If the recording is in progress, this method requests that the recording be stopped. If the recording is pending, this method requests that the recording be canceled. Upon completion of this operation, a NetActionEvent SHALL be delivered to the given handler indicating success or failure.

**Parameters:**

recording - a RecordingContentItem or NetRecordingEntry that identifies the recording(s) to be canceled.

handler - The NetActionHandler which gets informed once this request completes.

**Returns:**

NetActionRequest to inform calling application of results.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("recording")  
 java.lang.IllegalArgumentException - if the recording parameter is neither the  
 NetRecordingEntry with upnp:srsRecordScheduleID metadata entry nor the RecordingContentItem with  
 upnp:srsRecordTaskID metadata entry in its own MetadataNode

### requestDeleteService

NetActionRequest **requestDeleteService**(ContentEntry recording,  
 NetActionHandler handler)

Requests that content associated with a scheduled recording be deleted from storage. Upon completion of this operation, a NetActionEvent SHALL be delivered to the given handler indicating success or failure.

**Parameters:**

recording - a RecordingContentItem or NetRecordingEntry that identifies the recording(s) to be deleted.

handler - The NetActionHandler which gets informed once this request completes.

**Returns:**

NetActionRequest to inform calling application of results.

**Throws:**

java.lang.SecurityException - if the caller does not have HomeNetPermission("recording")

### requestDelete

NetActionRequest **requestDelete**(ContentEntry recording,  
 NetActionHandler handler)

Requests that metadata associated with a scheduled recording be deleted from storage. Upon completion of this operation, a NetActionEvent SHALL be delivered to the given handler indicating success or failure.

**Parameters:**

recording - a recording that identifies the recording to be deleted.

handler - The NetActionHandler which gets informed once this request completes.

**Returns:**

NetActionRequest to inform calling application of results.

**Throws:**

`java.lang.SecurityException` - if the caller does not have `HomeNetPermission("recording")`

**requestPrioritize**

`NetActionRequest requestPrioritize(RecordingContentItem[] recordings, NetActionHandler handler)`

Requests that a group of scheduled individual recordings be prioritized on this network recording device. Prioritization is determined by the ordering of recordings in the array of `RecordingContentItems`, with highest priority given to the entry at element 0 in the array. Upon completion of this operation, a `NetActionEvent` SHALL be delivered to the given handler indicating success or failure.

**Parameters:**

`recordings` - a prioritized array of `RecordingContentItems`

`handler` - The `NetActionHandler` which gets informed once this request completes.

**Returns:**

`NetActionRequest` to inform calling application of results.

**Throws:**

`java.lang.SecurityException` - if the caller does not have `HomeNetPermission("recording")`

**requestPrioritize**

`NetActionRequest requestPrioritize(NetRecordingEntry[] recordings, NetActionHandler handler)`

Requests that a group of scheduled recording request be prioritized on this network recording device, where each recording request may represent one or more individual recordings on the remote device. Prioritization is determined by the ordering of recordings in the array of `NetRecordingEntries` with highest priority given to the entry at element 0 in the array. Upon completion of this operation, a `NetActionEvent` SHALL be delivered to the given handler indicating success or failure.

**Parameters:**

`recordings` - a prioritized array of `NetRecordingEntries`

`handler` - The `NetActionHandler` which gets informed once this request completes.

**Returns:**

`NetActionRequest` to inform calling application of results.

**Throws:**

`java.lang.SecurityException` - if the caller does not have `HomeNetPermission("recording")`

## Annex G Security API

Package `org.ocap.hn.security`

### Interface Summary

<b>NetAuthorizationHandler</b>	This interface represents a callback mechanism to a registered network authorization handler application.
--------------------------------	---

### Class Summary

<b>NetSecurityManager</b>	This class provides access to home network security capabilities including password handling.
---------------------------	---

## org.ocap.hn.security Interface NetAuthorizationHandler

```
public interface NetAuthorizationHandler
```

This interface represents a callback mechanism to a registered network authorization handler application.

### Method Summary

boolean	<b>notifyAction</b> (java.lang.String actionName, java.net.InetAddress inetAddress, java.lang.String macAddress, int activityId) Notifies the authorization handler application that an action it registered interest in has been received.
void	<b>notifyActivityEnd</b> (int activityId) Notifies the registered authorization handler application that an activity has ended.
boolean	<b>notifyActivityStart</b> (java.net.InetAddress inetAddress, java.lang.String macAddress, java.net.URL url, int activityId) Notifies the registered authorization handler application that an activity to access cable services has been started.

### Method Detail

#### notifyActivityStart

```
boolean notifyActivityStart(java.net.InetAddress inetAddress,
                             java.lang.String macAddress,
                             java.net.URL url,
                             int activityId)
```

Notifies the registered authorization handler application that an activity to access cable services has been started. The handler application will accept or deny the ability for the activity to continue.

**Parameters:**

inetAddress - IP address the transaction was sent from.

macAddress - MAC address the transaction was sent from if present at any layer of the received communications protocol. Can be empty String if not present. The format is EUI-48 with 6 colon separated 2 digit bytes in hexadecimal notation with no leading "0x", e.g. "00:11:22:AA:BB:CC".

url - The URL requested by the transaction.

activityId - The unique identifier of the activity, e.g. recorded content playback.

**Returns:**

True if the activity is accepted, otherwise returns false.

#### notifyActivityEnd

```
void notifyActivityEnd(int activityId)
```

Notifies the registered authorization handler application that an activity has ended.

**Parameters:**

activityId - Unique identifier assigned to the activity and passed to the notifyActivityStart method.

**notifyAction**

```
boolean notifyAction(java.lang.String actionName,  
                      java.net.InetAddress inetAddress,  
                      java.lang.String macAddress,  
                      int activityId)
```

Notifies the authorization handler application that an action it registered interest in has been received.

**Parameters:**

`actionName` - Name of the action received. Will match the name passed to the

`NetSecurityManager.setAuthorizationApplication` method.

`inetAddress` - IP address the transaction was sent from.

`macAddress` - MAC address the transaction was sent from if present at any layer of the received communications protocol. Can be empty `String` if not present. The format is EUI-48 with 6 colon separated 2 digit bytes in hexadecimal notation with no leading "0x", e.g. "00:11:22:AA:BB:CC".

`activityId` - The unique identifier of the activity if known. If no `activityId` is association with the transaction the implementation SHALL pass a value of -1;

**Returns:**

True if the activity is accepted, otherwise returns false.

## org.ocap.hn.security Class NetSecurityManager

```
java.lang.Object
└─org.ocap.hn.security.NetSecurityManager
```

```
public abstract class NetSecurityManager
extends java.lang.Object
```

This class provides access to home network security capabilities including password handling. The passwords that can be handled are specific to each home network interface at the link layer. Upper layer (e.g. TLS) and Administrator passwords cannot be accessed using this class. When the network interface type returned by `NetworkInterface.getType()` is MOCA the implementation SHALL associate the `getNetworkPassword` and `setNetworkPassword` methods in this interface to the MoCA link layer password used for the network interface. When the network interface type returned by `NetworkInterface.getType()` is WIRELESS\_ETHERNET the implementation SHALL associate the `getNetworkPassword` and `setNetworkPassword` in this interface to the link layer password, e.g. WEP, used for the network interface.

### Constructor Summary

protected	<b>NetSecurityManager()</b> Protected constructor.
-----------	---

### Method Summary

void	<b>disableMocaPrivacy</b> (NetworkInterface networkInterface) Disables MoCA privacy.
void	<b>enableMocaPrivacy</b> (NetworkInterface networkInterface) Enables MoCA privacy.
static NetSecurityManager	<b>getInstance()</b> Get the network security manager.
java.lang.String	<b>getNetworkPassword</b> (NetworkInterface networkInterface) Gets a network interface password.
boolean	<b>queryTransaction</b> (java.lang.String actionName, java.net.InetAddress inetAddress, java.lang.String macAddress, java.net.URL url, int activityId) Queries the implementation to determine if it has sent a transaction matching the parameters.
void	<b>revokeAuthorization</b> (long activityID) Revokes a session authorization granted by the authorization handler.
void	<b>setAuthorizationHandler</b> (NetAuthorizationHandler nah) Sets the authorization handler application.

## Method Summary

void	<b>setAuthorizationHandler</b> (NetAuthorizationHandler nah, java.lang.String[] actionNames, boolean notifyTransportRequests) Sets the authorization handler application.
void	<b>setNetworkPassword</b> (NetworkInterface networkInterface, java.lang.String password) Sets a network interface password.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### NetSecurityManager

protected **NetSecurityManager**()  
Protected constructor.

## Method Detail

### getInstance

public static NetSecurityManager **getInstance**()  
Get the network security manager.

### getNetworkPassword

public java.lang.String **getNetworkPassword**(NetworkInterface networkInterface)  
Gets a network interface password.

#### Parameters:

networkInterface - The interface to get the password for.

#### Returns:

The value of the password requested, or 0 length String if no password is set for the interface. If the interface type is MoCA this method returns a string value equal to the corresponding mocaIfPassword MIB. In this case the password MAY have been set using means other than the setNetworkPassword method.

#### Throws:

java.lang.UnsupportedOperationException - if a password cannot be retrieved for the network interface.

java.lang.SecurityException - if the calling application has not been granted MonitorAppPermission("handler.homenetwork").

### setNetworkPassword

public void **setNetworkPassword**(NetworkInterface networkInterface, java.lang.String password)

Sets a network interface password. If the network interface type is MoCA then privacy must also be enabled for the password to have affect. See the enableMocaPrivacy method. If the interface type is MoCA and the parameter is acceptable this method writes the corresponding mocaIfPassword MIB.

**Parameters:**

`networkInterface` - The home network interface the password is to be set for.

`password` - The value of the password to set.

**Throws:**

`java.lang.IllegalArgumentException` - if the password format is invalid for the interface type. A password for a MoCA interface that is less than 12 characters or greater than 17 characters or has any non-numerical characters is invalid.

`java.lang.UnsupportedOperationException` - if a password cannot be set for the network interface.

`java.lang.SecurityException` - if the calling application has not been granted `MonitorAppPermission("handler.homenetwork")`.

**setAuthorizationHandler**

```
public void setAuthorizationHandler(NetAuthorizationHandler nah)
```

Sets the authorization handler application. If a handler application is already registered this method SHALL replace it. If the parameter is null any set handler is removed.

**Parameters:**

`nah` - Network authorization handler interface to the authorization application.

**Throws:**

`java.lang.SecurityException` - if the calling application is not granted `MonitorAppPermission("handler.network")`.

**setAuthorizationHandler**

```
public void setAuthorizationHandler(NetAuthorizationHandler nah,
                                     java.lang.String[] actionNames,
                                     boolean notifyTransportRequests)
```

Sets the authorization handler application. If a handler application is already registered this method SHALL replace it. If the `nah` parameter is null any set handler is removed.

**Parameters:**

`nah` - Network authorization handler interface to the authorization application.

`actionNames` - A list of action names the handler is interested in authorizing. The format of the names is out-of-scope for this definition.

`notifyTransportRequests` - A true value indicates the `NetAuthorizationHandler` application is always notified. A false value indicates that once a message in a session has been authorized subsequent messages in the same session are not authorized. That is, the `NetAuthorizationHandler` application is not notified when the subsequent messages are received.

**Throws:**

`java.lang.SecurityException` - if the calling application is not granted `MonitorAppPermission("handler.network")`.

`java.lang.IllegalArgumentException` - if the `actionNames` parameter contains a name that cannot be matched to a known action.

**revokeAuthorization**

```
public void revokeAuthorization(long activityID)
```

Revokes a session authorization granted by the authorization handler.

**Parameters:**

`activityID` - The activity specific identifier that was passed to the `NetAuthorizationHandler.notifyAuthorization` method.

**Throws:**

`java.lang.SecurityException` - if the calling application does not have `MonitorAppPermission("handler.network")`.

### queryTransaction

```
public boolean queryTransaction(java.lang.String actionName,  
                               java.net.InetAddress inetAddress,  
                               java.lang.String macAddress,  
                               java.net.URL url,  
                               int activityId)
```

Queries the implementation to determine if it has sent a transaction matching the parameters.

**Parameters:**

`actionName` - Name of the request type if known. If not known an empty string MAY be used. The format of the name is out-of-scope of this definition.

`inetAddress` - IP address the transaction was sent to.

`macAddress` - MAC address the transaction was sent from if known. Can be empty String if not known. The format is EUI-48 with 6 colon separated 2 digit bytes in hexadecimal notation with no leading "0x", e.g. "00:11:22:AA:BB:CC".

`url` - The URL requested by the transaction if known. If not known an empty string may be used.

`activityId` - The activity identifier this device set for the connection. A value of -1 indicates the parameter will not be used for transaction matching purposes.

**Returns:**

True if `activityId` and other known parameters can be matched to a transaction sent by the implementation. If `activityId` match cannot be found, or if `activityId` match is found but any of the other known parameters do not match the transaction then this method returns false.

**Throws:**

`java.lang.IllegalArgumentException` - if the MAC address is malformed.

`java.lang.SecurityException` - if the calling application is not signed.

### enableMocaPrivacy

```
public void enableMocaPrivacy(NetworkInterface networkInterface)
```

Enables MoCA privacy. For MoCA interface types this method enables privacy and writes the corresponding `mocalfPrivacyEnable` MIB with a value of 'true'.

**Parameters:**

`networkInterface` - Interface to enable privacy on.

**Throws:**

`java.lang.UnsupportedOperationException` - if the parameter interface is not a MoCA interface type.

`java.lang.SecurityException` - if the calling application has not been granted `MonitorAppPermission("handler.homenetwork")`.

### disableMocaPrivacy

```
public void disableMocaPrivacy(NetworkInterface networkInterface)
```

Disables MoCA privacy. For MoCA interface types this method disables privacy and writes the corresponding `mocalfPrivacyEnable` MIB with a value of 'false'.

**Parameters:**

`networkInterface` - Interface to disable privacy on.

**Throws:**

`java.lang.UnsupportedOperationException` - if the parameter interface is not a MoCA interface type.

`java.lang.SecurityException` - if the calling application has not been granted `MonitorAppPermission("handler.homenetwork")`.

## Appendix I      Revision History

The following ECNs were incorporated into OC-SP-OCAP-HNEXT-I02-071220:

<b>ECN</b>	<b>Date Accepted</b>	<b>Title of EC</b>
OCAP-HNEXT-N-05.0844-1	12/19/05	System property identifying Home Networking extension
OCAP-HNEXT-N-06.0859-1	2/14/06	HN Mapping
OCAP-HNEXT-N-06.0865-1	2/7/06	Correction of System property name identifying Home Networking extension
OCAP-HNEXT-N-07.1079-2	9/25/07	OCAP Home Networking usability cleanup

The following ECNs were incorporated into OC-SP-OCAP-HNEXT-I03-080418:

<b>ECN</b>	<b>Date Accepted</b>	<b>Title of EC</b>
OCAP-HNEXT-N-08.1164-6	4/1/08	OCAP Home Networking Extension Phase II Version 1
OCAP-HNEXT-N-08.1241-1	4/15/08	UPnP Constants additions

The following ECNs were incorporated into OC-SP-OCAP-HNEXT-I04-091217:

<b>ECN</b>	<b>Date Accepted</b>	<b>Title of EC</b>
OCAP-HNEXT-N-08.1253-1	7/14/08	Home Networking Device InetAddress
OCAP-HNEXT-N-08.1261-2	7/14/08	HN Overlapping Methods
OCAP-HNEXT-N-08.1280-1	12/17/09	HN Security API
OCAP-HNEXT-N-08.1302-5	12/17/09	HN Authorization API
OCAP-HNEXT-N-08.1306-4	12/17/09	Miscellaneous HNEXT Phase II cleanup
OCAP-HNEXT-N-08.1333-2	12/17/09	Clarify the behavior of the method of NetRecordingEntry
OCAP-HNEXT-N-08.1336-2	12/17/09	Clarify the behavior of the method of ContentContainer
OCAP-HNEXT-N-08.1337-2	12/17/09	Remove ContentManagementNetModule interface
OCAP-HNEXT-N-08.1338-1	12/17/09	Add and clarify the methods of RecordingContentItem
OCAP-HNEXT-N-08.1339-1	12/17/09	Remove ContentDatabase class from HNEXT

<b>ECN</b>	<b>Date Accepted</b>	<b>Title of EC</b>
OCAP-HNEXT-N-08.1344-1	12/17/09	Clarify SRS related methods in OCAP HNext
OCAP-HNEXT-N-08.1345-2	12/17/09	Remove a recording parameter from RecordingContentItem.requestSetMediaTime()
OCAP-HNEXT-N-08.1357-4	12/17/09	Home Networking Phase 2.0 API update
OCAP-HNEXT-N-08.1367-4	12/17/09	Add IllegalArgumentException conditions to RecordingNetModule
OCAP-HNEXT-N-09.1370-1	12/17/09	Modify action state definitions of NetActionEvent
OCAP-HNEXT-N-09.1384-2	12/17/09	Add Device.setProperties method
OCAP-HNEXT-N-09.1392-5	12/17/09	Remove HN Unchecked Exceptions
OCAP-HNEXT-N-09.1425-2	12/17/09	Update OCAP Reference
OCAP-HNEXT-N-09.1434-1	12/17/09	Corrections to make consistent use of scheduledStartDateTime
OCAP-HNEXT-N-09.1452-1	12/17/09	Fix inconsistencies in ObjectID handling between the APIs and UPnP behaviors
OCAP-HNEXT-N-09.1460-1	12/17/09	Fix AudioResource language APIs
OCAP-HNEXT-N-09.1462-1	12/17/09	Fix DatabaseQuery.or() throws clause
OCAP-HNEXT-N-09.1463-1	12/17/09	Correct Monitor App Permission Statement in 6.5.3
OCAP-HNEXT-N-09.1465-1	12/17/09	Clarify ContentContainer.deleteRecursive()
OCAP-HNEXT-N-09.1469-1	12/17/09	NetActionRequest.getError() clarification

The following ECNs were incorporated into OC-SP-OCAP-HNEXT-I05-100603:

<b>ECN</b>	<b>Date Accepted</b>	<b>Title of EC</b>
OCAP-HNEXT-N-09.1483-1	6/3/10	Remove Section 8
OCAP-HNEXT-N-10.1545-1	6/3/10	Home Networking Asset In Use Detection