Oracle® Fusion Middleware

Content Management REST Service Developer's Guide

11*g* Release 1 (11.1.1)

E15813-04

February 2010

1 Introduction

The OASIS CMIS (Content Management Interoperability Services) Technical Committee works to standardize a web services interface specification that will enable greater interoperability of Enterprise Content Management (ECM) systems. For more information, see the Oasis CMIS site: `http://www.oasis-open.org/committees/cmis/`

The Content Management REST Service provides a server that uses the CMIS RESTful AtomPub server binding to provide access to the CM VCR (Content Management Virtual Content Repository).

This guide is intended to be a supplement to the OASIS CMIS specification, and provides details on the specific implementation of the Content Management REST Service. Before continuing, all users should review the OASIS CMIS specification. This guide references the Public Review Draft of the Content Management Interoperability Services (CMIS) Version 1.0 (Committee Draft 04), which can be viewed at the following URL: `http://docs.oasis-open.org/cmis/CMIS/v1.0/cd04/cmis-spec-v1.0.html`. The specification includes the domain model and two server bindings. As mentioned above, only the RESTful AtomPub binding is currently implemented by the Content Management REST Service. The specification includes XML schemas (some common to both bindings and others not) and examples. Users should also be familiar with AtomPub, as this is the default formatting for the responses.

> **Note:** CMIS provides a lowest common denominator for a wide range of different content systems; it is not aligned directly with the Content Management REST Service. Not everything from the CM VCR is exposed by CMIS and not everything in CMIS is supported by the CM VCR.

This guide includes the following sections:

- Section 2, "CMIS Part I - Domain Model"
- Section 3, "CMIS Part II: RESTful AtomPub Binding"
- Section 4, "Content Management REST Service Best Practices and Examples"

2 CMIS Part I - Domain Model

The Domain Model part of the CMIS specification defines a domain model that can be used by applications to work with one or more Content Management repositories/systems.

This section is organized according to the sections in the CMIS Domain Model documentation.

- Section 2.1, "Data Model"
- Section 2.2, "Services"

2.1 Data Model

The Content Management REST Service service document consists of AtomPub workspaces. Each workspace maps to a content connection (only UCM repositories are supported by the Content Management REST Service). For details on the service document, see the next section, Section 3, "CMIS Part II: RESTful AtomPub Binding".

2.1.1 Repository

For this release, some of the optional capabilities listed in section 2.1.1 have not been implemented. Versioning, ACL, Policies, Relationships, Change Log, Folder Descendants/Tree, Renditions and Full Text Search will be considered for future releases.

Specifically, the Content Management REST Service implementation has the following optional capabilities:

```
capabilityGetDescendants = false
capabilityContentStreamUpdatability = anytime
capabilityChanges = none
capabilityRenditions = none
capabilityMultifiling = false
capabilityUnfiling = false
capabilityVersionSpecificFiling = false
capabilityPWCUpdateable = false
capabilityPWCSearchable = false
capabilityAllVersionsSearchable = false
capabilityJoin = none
capabilityACL = none
capabilityQuery = none or metadataonly
```

Note: Full text search is not currently implemented. `capabilityQuery` will be `metadataonly` if search is enabled in the repository configuration. Otherwise it will be `none`.

2.1.2 Object

Content Management REST Service supports document and folder objects. In CMIS the `cmis:baseTypeId` for a Node will be `cmis:folder` or `cmis:document`. Also, the `cmis:baseId` for a Type will be `cmis:folder` or `cmis:document`. A CMIS object's ID (`cmis:objectid`) is directly related to the CM VCR Node ID.

VCR property types are mapped to CMIS property-types as follows:

- Property.STRING: cmis:propertyString

- BOOLEAN: cmis:propertyBoolean
- DOUBLE: cmis:propertyDecimal
- LONG: cmis:propertyInteger
- CALENDAR: cmis:propertyDateTime
- NESTED: cmis:propertyId
- LINK: cmis:propertyId
- BINARY: (Not directly represented as a cmis type, but one binary property will be exposed as 3 properties: cmis:contentStreamLength, cmis:contentStreamMimeType and cmis:contentStreamId. Or with getContentStream/setContentStream.)

2.1.3 Object-Type

CMIS Object-Type aligns with the CM VCR ObjectClass, also referred to as Type.

2.1.4 Document Object

Document Objects are the elementary information entities managed by the repository. As defined by the CMIS specification, Document Objects may be version-able, file-able, query-able, control-able and ACLControl-able. As stated earlier, the Content Management REST Service does not support versioning, multi-filing, Policies or ACL for this release.

If a Node is determined to be a Document (not a Folder) then any children it has will not be exposed via CMIS. In CMIS, each Document Object can be associated with a single content stream. Therefore, the Content Management REST Service will expose a maximum of one content stream, even if more are associated with the Node.

2.1.5 Folder Object

The CMIS specification states that Folder Objects do not have a content-stream and are not version-able. If a Node is determined to be a Folder, then the Content Management REST Services exposes it in this manner. (In UCM, folders do not have a content stream and are not versionable).

2.1.6 Relationship Object

The Relationship Object section does not apply, since the Content Management REST Service does not support Relationships for this release.

2.1.7 Policy Object

The Policy Object section does not apply, since the Content Management REST Service does not support Policies for this release.

2.1.8 Access Control

Most of the Access Control section does not apply, since the Content Management REST Service does not support ACL for this release. See below for details on allowable actions.

2.1.8.1 AllowableActions Mapping This section lists allowable actions that will be defined for Objects. Because of how this release is implemented, some of these are hard-coded for all objects. Other allowable actions will be set based on the repository configuration.

- canGetObjectRelationships = false

- canCreateRelationship = false
- canGetDescendants = false
- canCheckOut = false (versioning)
- canCancelCheckOut = false (versioning)
- canCheckIn = false (versioning)
- canAddObjectToFolder = false (multi-filing)
- canRemoveObjectFromFolder = false (unfiling/multi-filing)
- canApplyPolicy = false
- canGetAppliedPolicies = false
- canRemovePolicy = false
- canCreatePolicy = false
- canApplyACL = false
- canGetACL = false
- canGetRenditions = false
- canDeleteTree = false
- canGetAllVersions = false (versioning)

2.1.9 Versioning

Section 2.10 does not apply, since the Content Management REST Service does not support versioning for this release.

2.1.10 Query

CMIS queries return a Result Set where each Entry object will contain only the properties that were specified in the query. As the Content Management REST Service does not support JOINs in queries, each result entry will represent properties from a single node. To most closely mimic the behavior of a CM VCR search, use a query like "SELECT * FROM …".

- The query is always focused on a single repository. Search.searchPath is always "/RepositoryId/".
- The FROM clause specifies an object-type to be searched. This translates into a VCR expression which is ANDed with the expression from the WHERE clause, so as to restrict the search to the desired type:
  - FROM cmis:document ==> cm_isContent == true
  - FROM cmis:folder ==> cm_isHierarchy == true
  - FROM typeQueryName ==> cm_objectClassInstance == type.name (when cm_objectClassInstance is a searchable system property, otherwise ==> cm_objectClass == type.name)
- The cmis:document type will be queryable if the cm_isContent system property is searchable.
- The cmis:folder type will be queryable if the cm_isHierarchy system property is searchable.

- Other types will be queryable if they are searchable in the repository (that is, ObjectClass.isSearchable returns true) AND one of cm_objectClass or cm_objectClassInstance is a searchable system property.
- A type's includedInSuperTypeQuery attribute will be true if cm_objectClassInstance is a searchable system property.
- The IN_FOLDER predicate is implemented as the VCR expression "cm_parent_uid == folder.uid".
- The IN_TREE predicate is implemented as the VCR expression "cm_path LIKE '/RepositoryId/folder/path/*'".
- Properties of cmis:document and cmis:folder will be queryable and orderable if their corresponding cm_ system property is searchable and sortable. The system property mappings are:
  - cmis:createdBy ==> cm_createdBy
  - cmis:lastModifiedBy ==> cm_modifiedBy
  - cmis:creationDate ==> cm_createdDate
  - cmis:lastModificationDate ==> cm_modifiedDate
  - cmis:name ==> cm_nodeName
  - cmis:contentStreamFileName ==> cm_binaryName
  - cmis:contentStreamLength ==> cm_binarySize
  - cmis:contentStreamMimeType ==> cm_contentType
  - cmis:objectId ==> cm_uid
  - cmis:objectTypeId ==> cm_objectClass

    Note: Because the cmis representation is unlike the VCR representation (ID vs. ObjectClass name), cmis:objectTypeId is never orderable.

  - cmis:path ==> cm_path
- Nested properties are not queryable or orderable.

  Note: Some repositories may have capabilities that are not representable in a CMIS query, and some repositories may have restrictions which will limit the CMIS-query predicates (or combinations of predicates) that can be used in a query. The above mappings should assist you in translating repository capabilities and restrictions into corresponding considerations for CMIS-queries.

For example queries, see Section 4, "Content Management REST Service Best Practices and Examples". For more information about UCM Search limitations and considerations in WebLogic Portal, see *Best Practices for Modeling Content* in the *WebLogic Portal UCM VCR Adapter Guide*.

2.2 Services

The methods described in the Services section are implemented by the Content Management REST Service; specific implementation details are covered in the next section, Section 3, "CMIS Part II: RESTful AtomPub Binding".

3 CMIS Part II: RESTful AtomPub Binding

The RESTful AtomPub Binding part of the CMIS specification defines a specification based on AtomPub that can be used by applications to work with one or more Content Management Repositories.

3.1 Configuration

The CMIS AtomPub services are configured in an application by referencing the application library module. Referencing the application module from the ear will install a web application to serve all CMIS requests. You might also want to override the `context-root` for the CMIS web application, which is by default `cmisrest`.

The application's META-INF/weblogic-application.xml will have a section like the following.

Note: The application library module for Oracle WebLogic Portal is content-management-cmis-rest-app-lib.ear.
The application library module for Oracle WebCenter is content-management-jps-cmis-rest-app-lib.ear.

```
----
  <library-ref>
       <library-name>content-management-jps-cmis-rest-app-lib</library-name>
  </library-ref>

  <library-context-root-override>
          <context-root>cmisrest</context-root>
          <override-value>myAppNameCMISrest</override-value>
  </library-context-root-override>
----
```

3.2 Service Document

All navigation of a repository begins with the AtomPub Service Document. From this document, all accessible content in a repository can be discovered via the collections, links, and templates.

The URI to the service document, relative to the CMIS web application's context-root, is /cmis/repository.

Therefore, if an application is deployed with a library-context-root-override as in the example above, the service document would be accessed via the following URL:

http://hostname:port/myAppNameCMISrest/cmis/repository

By default, this document will contain a workspace for each configured UCM repository (only UCM repositories are supported by the CM VCR in Oracle WebCenter). A service document for a single repository can be obtained by using the repositoryId query parameter, as described in section 5 of the CMIS AtomPub binding specification.

As noted in the previous section, the service document consists of AtomPub workspaces. Each workspace maps to a CM VCR repository.

Specific URIs beyond the service document are not published; it is assumed that users will start at the service document and navigate the collections and links down expected paths. The relationships of the links and the titles and types of the collections are all defined in the CMIS specification, and thus can be commonly navigated by a client implementation. There are also templates defined for each repository, for easier access of objects by path, object by ID, type by ID, and queries. The format of the variables for the path and ID templates can be discovered by viewing the Entries of Folders and Documents.

3.3 Response Formats

Section 2.2: Response Formats indicates that Atom/AtomPub style formats will be returned by default unless overridden by a supported media type expressed in the Accept header.

A generic AtomPub feed reader can walk through any of the feeds returned by the CMIS REST server. It will not see all the CMIS specifics, but will be able to navigate through links. In general, to set up a feed reader, you need to know the URI of a particular feed, which can be discovered by navigating through the service document, for example, the workspace link for "typesdescendants".

For details on query syntax, see the CMIS specification. For Content Management REST Service best practices and examples, see the next section, Section 4, "Content Management REST Service Best Practices and Examples".

4 Content Management REST Service Best Practices and Examples

This section provides best practices and examples using the Content Management REST Service. For details on query syntax, see the CMIS specification.

4.1 Best Practices

The following list provides suggested best practices for repositories that will use the Content Management REST Service.

- Make a clear distinction between folder and document Nodes.
- To keep queries more readable, avoid non-alphanumeric characters in ObjectType and PropertyDefinition names.

4.2 Content Management REST Service Examples

This section provides some example queries. For details on query syntax, see the CMIS specification. (To get the full URI for a query, see the query URI template in the service document.)

- SELECT * from cmis:folder
- SELECT cmis:name, cmis:contentStreamFileName, cmis:contentStreamMimeType, cmis:contentStreamLength FROM cmis:document WHERE

cmis:contentStreamFileName = 'BinaryName' AND cmis:contentStreamMimeType = 'text/html' AND cmis:contentStreamLength > 1

- SELECT cmis:name, cmis:creationDate, cmis:lastModificationDate FROM cmis:folder WHERE cmis:name = 'Trash' AND cmis:lastModificationDate > TIMESTAMP '2008-05-18T10:32:44.703-06:00'
- SELECT * FROM cmis:document WHERE cmis:name LIKE 'baker%'
- SELECT * FROM cmis:document WHERE cmis:name NOT IN ('nodeBoolean', 'nodeLong')
- SELECT cmis:name from cmis:document where IN_TREE('/StellentRepository')
- SELECT * FROM ora:t:IDC:GlobalProfile where ora:p:dRevClassID > 1
- SELECT * FROM ora:t:IDC:GlobalProfile where ora:p:xBooleanTestField = FALSE
- SELECT ora:p:xMultiValuedDelimiterTest from ora:t:IDC:GlobalProfile where ANY ora:p:xMultiValuedDelimiterTest NOT IN ('four')

5 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Oracle Fusion Middleware Content Management REST Service Developer's Guide, 11*g* Release 1 (11.1.1)
E15813-04

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.