



## OpenText™ Documentum™ Content Management

### **Foundation REST API Reference Guide**

Build REST web service APIs or clients.

EDCPKRST250400-PRE-EN-01

---

**OpenText™ Documentum™ Content Management  
Foundation REST API Reference Guide**  
EDCPKRST250400-PRE-EN-01  
Rev.: 2025-Oct-18

This documentation has been created for OpenText™ Documentum™ Content Management CE 25.4.  
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product,  
on an OpenText website, or by any other means.

**Open Text Corporation**

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111  
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440  
Fax: +1-519-888-0677  
Support: <https://support.opentext.com>  
For more information, visit <https://www.opentext.com>

**© 2025 Open Text**

Patents may cover this product, see <https://www.opentext.com/patents>.

**Disclaimer**

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However,  
Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the  
accuracy of this publication.

---

# Table of Contents

<b>1</b>	<b>Overview .....</b>	<b>7</b>
1.1	Hypermedia-driven API .....	8
<b>2</b>	<b>Resources .....</b>	<b>13</b>
2.1	Organization of Resource Reference Documentation .....	13
2.2	Installing Additional Services .....	14
2.3	All Versions .....	16
2.4	ACL .....	37
2.5	ACL Associations .....	60
2.6	Archived Contents .....	65
2.7	Aspect Type (s) .....	73
2.8	Audit Policy (ies) .....	81
2.9	Audit Trail (s) .....	95
2.10	Available Audit Events .....	106
2.11	Batches .....	110
2.12	Batch Capabilities .....	128
2.13	Cabinet (s) .....	132
2.14	Checked Out Objects .....	143
2.15	Checksum .....	149
2.16	Comment (s) .....	151
2.17	Comment Replies .....	163
2.18	Content (s) .....	171
2.19	Content Media .....	185
2.20	Current User .....	188
2.21	Current User Preference (s) .....	191
2.22	Current Version .....	205
2.23	Custom Facets .....	207
2.24	Document .....	214
2.25	DQL .....	224
2.26	Folder .....	233
2.27	Folder Child Documents .....	240
2.28	Folder Child Folders .....	268
2.29	Folder Child Link (s) .....	274
2.30	Folder Child Objects .....	282
2.31	Format (s) .....	302
2.32	Group (s) .....	308
2.33	Group User (s) .....	322
2.34	Get Combined Groups and Members .....	328
2.35	Home Document .....	333
2.36	Inbox .....	337

2.37	Inbox Item .....	343
2.38	Job(s) .....	353
2.39	Lifecycle (s) .....	370
2.40	Lightweight Objects .....	400
2.41	Lightweight Object Parent .....	408
2.42	Lightweight Object Materialization .....	413
2.43	Lock .....	416
2.44	Module(s) .....	420
2.45	Multiple Repository Search .....	432
2.46	Network Location (s) .....	446
2.47	Object Aspect (s) .....	452
2.48	Object Lifecycle .....	460
2.49	Object Lifecycle Demotion .....	469
2.50	Object Lifecycle Promotion .....	475
2.51	Object Lifecycle Resumption .....	481
2.52	Object Lifecycle Suspension .....	487
2.53	Parent Link (s) .....	493
2.54	Permissions .....	503
2.55	Permission Set .....	507
2.56	Product Information .....	517
2.57	Registered Audit Event (s) .....	520
2.58	Relation (s) .....	534
2.59	Related Documents .....	542
2.60	Related SME .....	550
2.61	Relation Type (s) .....	556
2.62	Repository .....	564
2.63	Repositories .....	573
2.64	Saved Search (es) .....	575
2.65	Saved Search Template (s) .....	611
2.66	Search .....	645
2.67	Significant Terms .....	670
2.68	Sub-Group (s) .....	675
2.69	Synonyms .....	682
2.70	Synonym Categories .....	686
2.71	SysObject .....	692
2.72	Task Package (s) .....	708
2.73	Task Package Documents .....	723
2.74	Task Package Note .....	725
2.75	Type (s) .....	727
2.76	User (s) .....	766
2.77	User Default Folder .....	780
2.78	User Permission Set .....	784

2.79	User Subscription (s) .....	787
2.80	Value Assistances .....	802
2.81	Virtual Document Node (s) .....	806
2.82	Virtual Document Snapshot(s) .....	833
2.83	Virtual Document Snapshot Nodes .....	849
2.84	Virtual Document Snapshot Frozen .....	854
2.85	Word Suggestions .....	863
2.86	Work Item(s) .....	869
2.87	Work Queue(s) .....	879
2.88	Work Queue Task(s) .....	887
2.89	Workflow(s) .....	906
2.90	Workflow Activity (ies) .....	935
2.91	Workflow Attachment(s) .....	950
2.92	Workflow Template(s) .....	959
2.93	Workspace Type Definition .....	975
2.94	Workspace Folder Template .....	1001
2.95	Workspace Instance .....	1026
<b>A</b>	<b>Link Relations .....</b>	<b>1043</b>
<b>B</b>	<b>REST Common Definition - URI Request Query Parameters .....</b>	<b>1047</b>
<b>C</b>	<b>REST Common Definition - HTTP Status Codes .....</b>	<b>1053</b>
<b>D</b>	<b>REST Common Definition - HTTP Headers .....</b>	<b>1055</b>
<b>E</b>	<b>AQL .....</b>	<b>1059</b>
E.1	Expressions in AQL .....	1064
E.2	Facet in AQL .....	1071
<b>F</b>	<b>Events Audited by Default .....</b>	<b>1077</b>



# Chapter 1

## Overview

This guide is the resource reference that is shipped with OpenText™ Documentum™ Content Management Foundation REST API (also known as the core resources). It is intended for software developers and architects who are building REST web services APIs or clients that interact with OpenText Documentum CM. The guide provides information on how to use each resource and their inherent operations. For general information about OpenText™ Documentum™ Content Management Foundation REST API, including how to develop your own REST resources, see *OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)*. Experimental API documentation, which is written in RADL, is available to users in both the SDK and the REST WAR file.

OpenText Documentum Content Management (CM) Foundation REST API is a set of RESTful web service interfaces that interact with OpenText Documentum CM.

Being developed in a purely RESTful style, Foundation REST API is hypertext-driven, server-side stateless, and content negotiable. This provides you with high efficiency and simplicity in programming, and it also makes all services easy to consume. These advantages make Foundation REST API the optimal choice for Web 2.0 applications and mobile applications to interact with OpenText Documentum CM repositories.

Foundation REST API models objects in OpenText Documentum CM repositories as resources and identifies resources by Uniform Resource Identifiers (URIs). It defines specific media types to represent resources and drives application state transfers by using link relations. Foundation REST API uses a number of HTTP standard methods (GET, PUT, POST, and DELETE) to manipulate resources over the HTTP protocol.

This guide is designed as a resource catalog that focuses on how resources are related to one another. It gives you a general description of each resource and also discusses the operations that you can perform on each resource.

## 1.1 Hypermedia-driven API

Foundation REST API is a purely hypermedia-driven REST API. The physical Uniform Resource Identifiers (URI) used to access the service's resources are discoverable at runtime by using link relations instead of using hard-coded values. Hypermedia is an important aspect of REST because it allows you to build services where clients and servers are decoupled. This allows those clients and servers to evolve independently of each other.

Before consuming Foundation REST API, a REST client requires the URI of the service's Home document. This Home document is the entry point for the whole REST service. The **Home document** URI is the only permanent URI path that the REST client must bookmark: For example `/services`.

Issuing an HTTP GET Request for the entry point returns the Home Document, which contains the link relations for all of the top-level resources that are exposed by Foundation REST API. By default, the resources exposed include the **Repositories** collection and the **Product Information** resource.

### Example 1-1: The Home document

Here is an example of a default Home Document:

```
{  
  resources: {  
    http://identifiers.emc.com/linkrel/repositories: {  
      href: "http://localhost:8080/dctm-rest/repositories"  
      hints: {  
        allow:  
        "GET"  
        representations:  
        "application/xml"  
        "application/json"  
        "application/atom+xml"  
        "application/vnd.emc.documentum+json"  
      }  
    }  
    about: {  
      href: "http://localhost:8080/dctm-rest/product-info"  
      hints: {  
        allow:  
        "GET"  
        representations:  
        "application/xml"  
        "application/json"  
        "application/vnd.emc.documentum+xml"  
        "application/vnd.emc.documentum+json"  
      }  
    }  
  }  
}
```

The key for each resource is an immutable link relation name that defines the relation to the target resource. Each resource has an `href` attribute that contains the URI of the resource.



**Note:** An `href` does not have to be the same from one REST API version to another and may change without notice. This underscores the importance of avoiding hard-coded URIs in your client applications.

Each link relation name, such as `self`, `edit`, and `about`, provides the client with unchangeable identifiers that identify specific resources.

#### ➡ Example 1-2: Delete Request

This code sample illustrates the `delete` link relation in the `rel` element. It also demonstrates that the purpose of the `href` link is to delete the current resource, which in this case is `acme02`.

```
"links": [
  {
    "rel": "http://identifiers.emc.com/linkrel/delete",
    "href": "http://localhost:8080/dctm-rest/repositories/acme02"
  }
]
```



The logical relations between OpenText Documentum CM Core resources are managed using either link relations or feed-to-entry containership. The following diagram lists OpenText Documentum CM Core resources from a logical relation point of view.



**Figure 1-1: Core Resources in Foundation REST API**

As you can see in the diagram above, the **Home Document** resource is the starting point for everything. For example, to obtain the representation of the **Repository** resource, you must follow these steps:

1. Issue a GET Request for the **Home Document** to discover the **Repositories** collection.
2. Issue a GET Request for the **Repositories** collection, which returns all of the links pointing to single **Repository** resources.
3. Search the returned links for one that points to the **Repository** resource that you want and use to perform your task.

The link relations in the **Repository** resource give you information about what operations you can perform on that resource. This approach determines the way that a hypermedia RESTful web service interacts with REST clients.



**Note:** The above diagram provides a visual overview of OpenText Documentum CM Core resources and their relationships. It is not intended to be used as a reference to all relationships between OpenText Documentum CM Core resources. For example, a **Document** resource contains a link relation to the **type** resource, but this relationship does not appear in the diagram. Furthermore, it does not mean that you must traverse the path shown in the diagram to get to a resource. For example, you can also get to a **SysObject** resource from the **DQL Query** resource.

The *OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)* provides you with detailed examples on how hypermedia links work in resource discovery and operation execution.



# Chapter 2

## Resources

### 2.1 Organization of Resource Reference Documentation

The following guidelines apply to each resource reference documentation entry:

- All method parameters are optional unless otherwise noted.
- The `DELETE` and `GET` methods do not have a request body.

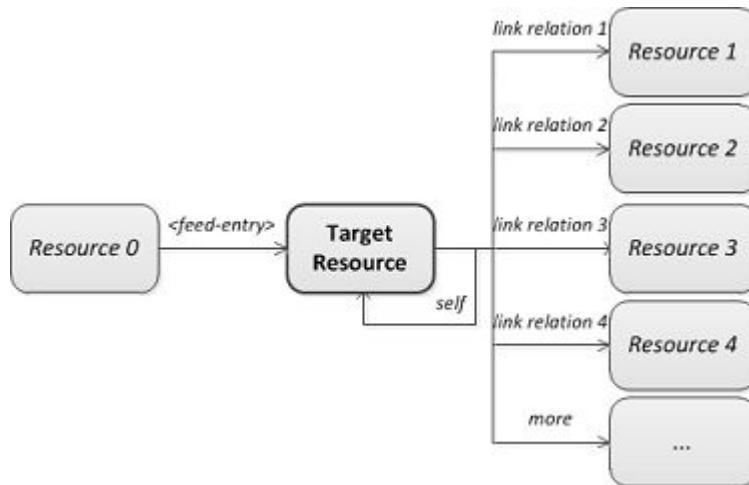
*About the resource URIs:*

Resource URIs are used to identify the physical locations of server resources. According to the best practices, resource URIs should be discovered using hypermedia, such as link relations and the response `Location` header. The only exception is the root entry URI of the Home Document resource.

You are encouraged to use hypermedia to discover your OpenText Documentum CM resources from the client side. Using hypermedia is far better than attempting to construct the URI in your client code. To help you with your REST client programming, see OpenText Documentum CM client samples available on the GitHub website.

*About the diagrams in resource relationships*

- The target resource is shown in bold.
- Link relationships are shown as strings on arrows.
- The resource to the left of the target resource (resource 0 in the sample) is the resource that you can use to navigate to the target resource using the link relation presented on the arrow. There can be more than one resource you can use to navigate to the target resource. However, only one resource is shown.
- Resources to the right of the target resource (resources 1 to 4 in the sample) are those that you can reach from the target resource. The diagrams displays at most four resources to the right of the target resource. Other resources, if they exist, are shown as suspension points such as (...).
- The string labelled as `more` on arrows is not a link relation. It is an indication that you can reach more than four resources from the target resource.
- The string `<feed-entry>` on arrows is not a link relation. It is an indication that two resources are linked using feed-to-entry containership.



**Figure 2-1: Resource relationships sample**



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

## 2.2 Installing Additional Services

Most of Foundation REST API can be used out-of-the-box when the OpenText™ Documentum™ Content Management Server repositories and the REST server are installed. However, the following services require the installation of additional services on OpenText Documentum Content Management (CM) Server:

- Collaboration services
- Search services
- Contextual Content Engine services
- OpenText™ Documentum™ Content Management Workflow Designer

### 2.2.1 Collaboration services

The commenting function is dependent on the Collaboration Services module. Therefore, the Collaboration Services Archive (DAR) files must be installed in the repository before using the following resources:

- “Comment” on page 151
- “Comments” on page 156
- “Comment Replies” on page 163

## 2.2.2 Search services

The full-text search function is dependent on the deployment and configuration of full-text indexing on OpenText Documentum CM repositories. The following resources are only available when full-text indexing is enabled on the target repositories:

- “Search” on page 645
- “Saved Searches” on page 586
- “Saved Search” on page 575
- “Saved Search Execution” on page 596
- “Saved Search Templates” on page 621
- “Saved Search Template” on page 611
- “Search Template Execution” on page 637
- DQL with full-text query using the parameter q with “Get versions” on page 17, “Get checked out objects” on page 145, “Get child objects from a folder” on page 283, and “Get all documents of a folder” on page 242
- “Custom Facets” on page 207

## 2.2.3 Contextual Content Engine services

The Contextual Content Engine services is a content recommendation system that adds contextual intelligence to Documentum CM Server. Before using the following CCE related services, a CCE instance must be installed and its URL and repository configured in the REST runtime. The following are the CCE related resources:

- “Related SME” on page 550
- “Related Documents” on page 542
- “Significant Terms” on page 670
- “Synonyms” on page 682
- “Synonym Categories” on page 686
- “Word Suggestions” on page 863

For detailed description about Contextual Content Engine related runtime property configuration, see to the Contextual Content Engine Service Configuration section of the `rest-api-runtime.properties.template` file in the `dctm-rest.war\WEB-INF\classes` folder.

## 2.2.4 Creating and managing workflow templates

To create and manage new workflow templates, use OpenText Documentum Content Management (CM) Workflow Designer. Use that application's Help for instructions on creating and managing the new workflow templates.

The following resources are only available when the OpenText™ Documentum™ Content Management Process Engine is enabled on the target repositories:

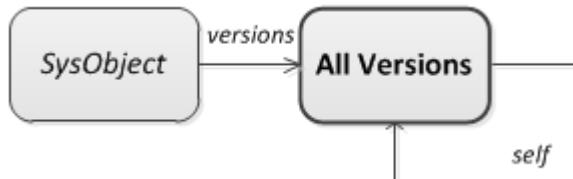
- Work Queue
- Work Queues
- Work Queue Task
- Work Queue Tasks

## 2.3 All Versions

The All Versions resource represents a collection of all versions for a given non-folder SysObject.

### 2.3.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



See Also: “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

### 2.3.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the versions
- Feed Title: All versions
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: Each entry represents a specific version of the given SysObject
- Entry ID: URI of the version
- Entry Title: object\_name for the specific version of the object

- Entry Updated: `r_modify_date` for the specific version of the object

### 2.3.3 Link relations

Link relation	Description	Resource reference
self	This All Versions resource	<a href="#">"All Versions" on page 16</a>
first, last, next, previous	Pagination links	<a href="#">"All Versions" on page 16</a>

### 2.3.4 Operations

#### 2.3.4.1 Supported HTTP methods

Method	Description
GET	Retrieves all versions of a given SysObject.
POST	Checks in a given >SysObject.

#### 2.3.4.2 Get versions

Retrieve all versions of a SysObject.

##### 2.3.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	version-label	Specifies the versions with the specified version label only.  A request can contain multiple version-label variables. Objects whose version-label matches any of those are returned.	string	N/A	null
	latest-versions-only	Specifies whether or not to show the latest versions only.	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	q	<p>Specifies the search criterion with a full-text expression in Simple Search Language.</p> <p>You must encode parameter q because it may contain non-English locale characters.</p> <p> <b>Note:</b> International characters that are used in this query parameter must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.</p>	string	N/A	No search criteria is used for the search.

This method also supports the following common query parameters:

- inline
- sort
- page

- items-per-page
- include-total
- view
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### **2.3.4.2.2 Request media types**

None

#### **2.3.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.3.4.2.4 Request body**

None

#### **2.3.4.2.5 Response headers**

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.3.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.3.4.2.7 Response statuses**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.3.4.2.8 Response body**

XML or JSON representation of all or a subset of versions of a given SysObject.

- Each object may contain all or a set of properties of the versions, depending on the setting of the `view` query parameter.
- The returned child objects collection only contains those items to which you have access.
- Pagination is supported.
- By default, the results are sorted by the `i_latest_flag` and `creation-date` properties in descending order.

### **2.3.4.3 Check in an object**

Check in the SysObject resource that has been checked out.

#### **2.3.4.3.1 HTTP Request method and query parameters**

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	content-count	<p>Specifies how many pieces of content are submitted. The metadata is excluded from the count.</p> <p>When more than one piece of content exists, the total content count must be provided. Otherwise, only the first content type is used. For example, when you import an object with one object metadata and five binary content types, then the value of the content-count parameter is 5.</p>	integer	>=0	0

HTTP method	Query parameters	Description	Data type	Value range	Default value
	all-primary	Specifies whether to import all contents using the same type as the primary content type for different pages (page 0, 1, 2...), or one primary content type, and others as renditions for the first page, which is page 0.	string	true or <i>false</i>	<i>true</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-length	<p>Specifies the byte count of the contents to be uploaded.</p> <p>This parameter is optional.</p> <p>A comma separated list can be used to specify multiple values for this parameter. For example: content-length=1024,2048,,4096. The blank value shown above is 0.</p> <p>When the all-primary parameter is <i>false</i>, the first value in the comma separated list is used for the content-length of the primary content, and all other values in the list are ignored.</p>	string	Comma separated string of integer number values	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	version-policy	<p>Specifies the version policy by using one of the following valid values:</p> <ul style="list-style-type: none"> <li>• next-major: The object receives the next major version number.</li> <li>• next-minor: The object receives the next minor version number.</li> <li>• branch-version: The object receives a branch version number.</li> <li>• none: No version policy is specified. Documentum CM Server makes the decision.</li> </ul>	string	A string containing any one of the valid values: next-major, next-minor, branch-version, or none.	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	retain-lock	<p>Specifies whether to keep the object checked out and locked by the user after the new version is saved.</p> <ul style="list-style-type: none"><li>• <i>true</i>: Keep the lock after the new version is saved.</li><li>• <i>false</i>: The object is not locked after the new version is saved.</li></ul>	string	true or <i>false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	version-label * on page 32	<p>Specifies one or more version labels of the checked in object with the following constraints:</p> <ul style="list-style-type: none"> <li>• By default, numeric version labels, such as 1.1, 2.1.0.1, 5.0 are not allowed.</li> <li>• You cannot set this parameter to CURRENT. To make this check-in the current version, use the make-current operation.</li> </ul>	string	One or more version labels of the checked in object	null
	comment	Specifies the comment for a check-in operation.	string	Any valid string	null
	page	Specifies the content page number with any non-negative number adhering to the constraints defined by Documentum CM Server.	integer	>=0	0

HTTP method	Query parameters	Description	Data type	Value range	Default value
	format	<p>Specifies the format when checking in more than one type of content. A comma separated string can be used to specify multiple values for the format parameter.</p> <p>For example:  <code>format=crtext,html,pub_html.</code></p> <p>The format list must match the content sequence. When you don't want to specify the format for a piece of content you can leave it blank in the comma separated list but you must still include the comma as a placeholder.</p> <p>For example:  <code>format=crtext,,html</code></p> <p>When the format for a piece of content is not defined, the server uses</p>	string	Comma separated string	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
		<p>the content media type to select the format.</p> <p>When the <code>all-primary</code> parameter is <code>true</code> and <code>page=0</code>, all primary content types must have the same format. In this case, when you define more than one format, they must be the same.</p> <p>For example: <code>format=crtext, crtext, crtext.</code></p> <p>When the <code>all-primary</code> parameter is <code>true</code> and the <code>page</code> parameter is greater than 0, you can define one format for all of the primary content. For example: <code>format=crtext.</code> In this case, all format values must be the same as the format of the primary</p>			

HTTP method	Query parameters	Description	Data type	Value range	Default value
		content of the first page, which is page 0.			
	make-current	Determines whether or not to make this check-in the current version.	string	true or <i>false</i>	<i>true</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-charset	<p>This parameter defines the parse behavior of the server when sanitizing content. A comma separated string can be used to specify multiple values for this parameter. For example: content-charset=UTF-8,GBK,ISO-8859-1</p> <p>When the content-charset parameter is not set and content is in HTML format, the server tries to get a value for the content-charset parameter from the metadata information of the content itself.</p> <p>When the metadata has a valid charset, then it is used to sanitize the content. Otherwise,</p>	string	Comma separated string of integer number values	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
		<p>the configuration default charset is used for the content-charset parameter when a value for this parameter and the metadata information are not found.</p> <p> <b>Note:</b> When the charset and content do not match, the uploaded content may have an incorrectly encoded value.</p>			

HTTP method	Query parameters	Description	Data type	Value range	Default value
	modifier	<p>Specifies the modifier of the contents. For the primary content, the modifier is ignored.</p> <p>For example, when you import with the all-primary parameter set to <i>false</i>, the value of the modifier can be: <code>modifier =,mod1, mod2,mod3 .</code></p> <p>That means the first (or primary) content type has no modifier, and the second rendition's modifier is mod1, the third is mod2 and so on.</p> <p>Setting the all-primary parameter to <i>true</i>, or when <code>content-count&lt;=1</code> causes all modifiers to be ignored.</p>	string	Comma separated string	null

\* Repeatable parameter



**Note:** Here are some additional details about the variables listed above:

- `format`

The `format` list must match the content sequence. When you don't want to specify the format for a piece of content you can leave it blank in the comma separated list but you must still include the comma as a placeholder.

For example: `format=crtext,,html`

When the format for contents is not defined, the server uses the content media type to select the format.

When the `all-primary` parameter is `true`, all primary content types must have the same format. In this case, when you define more than one format, they must be the same

For example: `format=crtext,crtext,crtext`. You can also define one format for all of the primary content: `format=crtext`

- `content-charset`

When the metadata has a valid charset, then it is used to sanitize the content.

When the `content-charset` is not provided and the content is in HTML format, the server tries to get a value for the `content-charset` parameter from the metadata information of the content itself.

The configuration default charset is used when the `content-charset` parameter and the metadata information are not found

When the charset and the content do not match, the uploaded content may have an incorrectly encoded value.

- `content-length`

When the `all-primary` parameter is set to `false`, the first value of in the `content-length` parameter is used for the primary piece of content, the other values in the comma separated string are ignored.

When storing content to Centera storage, the `content-length` parameter is mandatory and must be provided.

For other storage mediums, such as harddisk, the `content-length` parameter is optional.

In all cases, including when the `content-length` parameter is not required, if a value for it is set, the value must be an accurate byte count of the contents to be uploaded.

- `content-count`

When you import more than one piece of contents, the REST server must know how many content pieces there are in total. The total number of content pieces cannot be obtained until the end of the multipart stream. Therefore, the client must specify this number in the value of the `content-count` parameter.

When the value set in the `content-count` parameter is greater than the actual number of content pieces, an exception is thrown because the system expects more content pieces.

When the value set in the `content-count` parameter is less than the actual number of content pieces, all content pieces that exist in addition to the number of pieces set in the `content-count` parameter are ignored.

The value provided by the `content-count` must be accurate.

#### **2.3.4.3.2 Request media types**

When the client tries to check in both the metadata and one or more pieces of content, the `Content-Type` must be one of the following:

- `multipart/form-data`
- `multipart/mixed`

In addition to the above, the `Content-Type` of the first part must either:

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

When the client tries to check in more than one piece of content without metadata, the `Content-Type` must be one of the following:

- `multipart/form-data`
- `multipart/mixed`

When the client tries to check in the metadata only, the `Content-Type` must be one of the following:

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

For content types other than any of the types mentioned above, the server treats the operation as a single `content-only` check-in.

#### **2.3.4.3.3 Request headers**

- `Authorization`
- `Accept`
- `Content-Type`

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.3.4.3.4 Request body

The Request body contains one of or both of the following:

- The object metadata in XML or JSON
- The pieces of binary content

##### 2.3.4.3.4.1 Request payload samples

###### ➡ Example 2-1: Metadata and binary content in XML

```
Content-Type: multipart/form-data; boundary=314159265358979
--314159265358979
Content-Type: application/vnd.emc.documentum+xml
<document xsi:type="dm_document"
    definition="http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/
    acme01/types/dm_document">
    <properties xsi:type="dm_document-properties">
        <object_name>Blank PowerPoint Presentation</object_name>
        ...
    </properties>
</document>

--314159265358979
Content-Disposition: form-data; name=binary1
Content-Type: text/plain

This is content
--314159265358979--
```



###### ➡ Example 2-2: Check in metadata and binary content in XML

```
--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+xml

<object>
    <properties>
        <object_name>xyz</object_name>
    </properties>
</object>

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the first content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the second content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```

```
This is the third content part
```

```
--314159265358979--
```



#### **Example 2-3: Check in metadata and binary content in JSON**

```
--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+json
{
  "properties": {
    "object_name": "xyz"
  }
}
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```

```
This is the first content part
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```

```
This is the second content part
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```

```
This is the third content part
```

```
--314159265358979--
```



#### **2.3.4.3.4.2 Request payload samples**



#### **Example 2-4: Check in binary content without metadata**

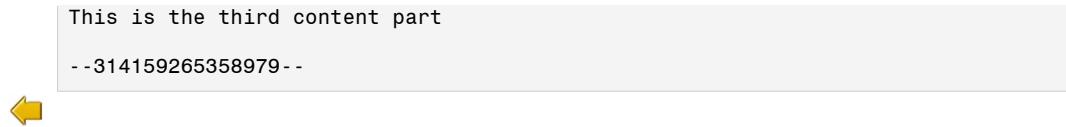
```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```

```
This is the first content part
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```

```
This is the second content part
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain
```



#### 2.3.4.3.5 Response headers

- Content-Type
- Location

#### 2.3.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.3.4.3.7 Response status

- 201 - Checked in successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.3.4.3.8 Response body

This is an XML or JSON representation of the object instance. The successful response contains all properties of the checked in object. A permission set is not returned.

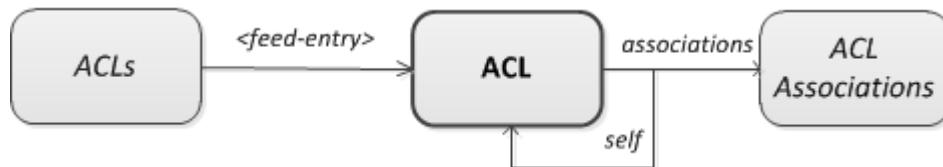
## 2.4 ACL

### 2.4.1 ACL

The ACL resource represents the object instance for an Access Control List (ACL).

### 2.4.1.1 Resource relationships

The following diagram shows how the ACL resource is related to other resources:



*See Also:* “[ACLs Collection](#)” on page 50 and [About the Diagram](#) on page 13.

### 2.4.1.2 Feed

No feed support.

### 2.4.1.3 Link relations

Link relation	Description	Resource reference
self	This ACL resource	<a href="#">“ACL” on page 37</a>
edit	Edit this ACL resource	<a href="#">“ACL” on page 37</a>
delete [1]	Delete this ACL resource	<a href="#">“ACL” on page 37</a>
associations [1]	Retrieve the sysobjects associated with the ACL	<a href="#">“ACL Associations” on page 60</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
`http://identifiers.emc.com/linkrel/`.

### 2.4.1.4 Operations

#### 2.4.1.4.1 Supported HTTP methods

Method	Description
POST	Updates the attributes of the ACL object instance
GET	Retrieves the attributes and other information of the ACL object instance
DELETE	Deletes the ACL object instance from the repository

#### 2.4.1.4.2 Get an ACL object instance

Retrieves the attributes and other information of the ACL object instance. Attributes are returned as embedded elements in the response message body. Other information is available within the link relations of the response message body.

##### 2.4.1.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- view
- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

##### 2.4.1.4.2.2 Request media types

N/A

##### 2.4.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

##### 2.4.1.4.2.4 Request body

N/A

##### 2.4.1.4.2.5 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.4.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.4.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.4.1.4.2.8 Response body

The properties block contains the following two sets of data:

- The metadata of this ACL
- ACL entries.

ACL entries are not presented one by one. Instead, an ACL resource displays the details of the entries in the following six elements:

- r\_accessor\_name
- r\_accessor\_permit
- r\_accessor\_xpermit
- r\_is\_group
- r\_permit\_type
- r\_application\_permit

These six elements contain the same number of items, which is also the number of entries in the ACL. The combination of items and elements positioning creates a key value pair that represents one ACL entry.

In the following code sample, the combination of all first items indicates that the accessor `<dm_world>`, which is an individual user where `<r_is_group>` is `false`, has Read basic permission (`<r_accessor_permit>` is 3), as well as EXECUTE\_PROCEDURE and CHANGE\_LOCATION extended permissions (`<r_accessor_xpermit>` is 0) on the objects to which this ACL is attached. It also indicates that the entry's permit type is `<AccessPermit>` (`r_permit_type` is 0).

#### Example 2-5: XML response

Status Code: 200 OK

```

Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<?xml version='1.0' encoding='UTF-8'?>
<acl
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="dm_acl"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_acl">
    <properties>
        <object_name>dm_4500000580000d07</object_name>
        <description>dm_4500000580000d07</description>
        <owner_name>dave</owner_name>
        <r_is_internal>true</r_is_internal>
        <r_accessor_name>
            <item>dm_world</item>
            <item>dm_owner</item>
            <item>dm_group</item>
        </r_accessor_name>
        <r_accessor_permit>
            <item>3</item>
            <item>7</item>
            <item>5</item>
        </r_accessor_permit>
        <r_accessor_xpermit>
            <item>0</item>
            <item>0</item>
            <item>3</item>
        </r_accessor_xpermit>
        <r_is_group>
            <item>false</item>
            <item>false</item>
            <item>false</item>
        </r_is_group>
        <globally_managed>false</globally_managed>
        <acl_class>0</acl_class>
        <r_has_events>false</r_has_events>
        <r_permit_type>
            <item>0</item>
            <item>0</item>
            <item>0</item>
        </r_permit_type>
        <r_application_permit>
            <item></item>
            <item></item>
            <item></item>
        </r_application_permit>
        <i_has_required_groups>false</i_has_required_groups>
        <i_has_required_group_set>false</i_has_required_group_set>
        <i_has_access_restrictions>false</i_has_access_restrictions>
        <r_template_id>0000000000000000</r_template_id>
        <r_alias_set_id>0000000000000000</r_alias_set_id>
        <i_partition></i_partition>
        <i_is_replica>false</i_is_replica>
        <i_vstamp>0</i_vstamp>
        <r_object_id>4500000580000d07</r_object_id>
    </properties>
    <links>
        <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/acls/4500000580000d07"/>
        <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/acls/4500000580000d07"/>
        <link rel="http://identifiers.emc.com/linkrel/associations">
    </links>

```

```
        href="http://localhost:8080/dctm-rest/repositories/
    REPO/acls/4500000580000d07/associations" />
<link rel="http://identifiers.emc.com/linkrel/delete"
    href="http://localhost:8080/dctm-rest/repositories/
    REPO/acls/4500000580000d07" />
</links>
</acl>
```



### ➤ Example 2-6: JSON response

Status Code: 200 OK

Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

```
{
    "name": "acl",
    "type": "dm_acl",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_acl",
    "properties": {
        "object_name": "dm_4500000580000d07",
        "description": "dm_4500000580000d07",
        "owner_name": "dave",
        "r_is_internal": true,
        "r_accessor_name": [
            "dm_world",
            "dm_owner",
            "dm_group"
        ],
        "r_accessor_permit": [
            3,
            7,
            5
        ],
        "r_accessor_xpermit": [
            0,
            0,
            3
        ],
        "r_is_group": [
            false,
            false,
            false
        ],
        "globally_managed": false,
        "acl_class": 0,
        "r_has_events": false,
        "r_permit_type": [
            0,
            0,
            0
        ],
        "r_application_permit": [
            "",
            "",
            ""
        ],
        "i_has_required_groups": false,
        "i_has_required_group_set": false,
        "i_is_group": false
    }
}
```

```

        "i_has_access_restrictions": false,
        "r_template_id": "0000000000000000",
        "r_alias_set_id": "0000000000000000",
        "i_partition": 0,
        "i_is_replica": false,
        "i_vstamp": 0,
        "r_object_id": "4500000580000d07"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580000d07"
        },
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580000d07"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580000d07"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/associations",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580000d07/
            associations"
        }
    ]
}

```



#### 2.4.1.4.3 Update the ACL object instance

Updates the ACL with specified attributes. When a client provides a request body that contains undefined or read only attributes, the server throws an exception. An ACL object can only be modified by its owner or the SuperUser.

The server throws an exception when the update request provides attributes whose values do not have matching positions in the repeating attributes, such as `r_accessor_name` and `r_accessor_permit`.

##### 2.4.1.4.3.1 HTTP Request method and query parameters

POST

#### 2.4.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.4.1.4.3.3 Request headers

- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.4.1.4.3.4 Request body

A single XML or JSON object instance. Only the following properties can be put into the request message body:

- <description>
- <acl\_class>
- <r\_accessor\_name>
- <r\_accessor\_permit>
- <r\_accessor\_xpermit>
- <r\_application\_permit>
- <r\_permit\_type>



#### Notes

- Except for <description> and <acl\_class>, all editable properties are prefixed with <r>, and must contain the same number of items. Otherwise, the REST server throws an exception.
- Every ACL has two built-in accessors: <dm\_world> and <dm\_owner>. The default permission value of these two accessors is defined by the Documentum CM Server configuration.
- To update an ACL template instance (acl\_class=2), you must change it to a regular (acl\_class=0) ACL, public (acl\_class=3) ACL, or template (acl\_class=1) ACL first. Additionally, you cannot change a regular, public, or template ACL into an ACL template instance.



#### Example 2-7: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<acl xmlns="http://identifiers.emc.com/vocab/documentum"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xsi:type="dm_acl">
    <properties>
      <r_accessor_name>
        <item>{accessor-name_1}</item>
```

```

        <item>{accessor-name_2}</item>
    </r_accessor_name>
    <r_accessor_permit>
        <item>{accessor-permit_1}</item>
        <item>{accessor-permit_1}</item>
    </r_accessor_permit>
    <r_permit_type>
        <item>{permit-type_1}</item>
        <item>{permit-type_1}</item>
    </r_permit_type>
    ...
</properties>
</acl>
```



#### Example 2-8: JSON request

```
{
    "name": "acl",
    "type": "dm_acl",
    "properties": {
        "r_accessor_name": [
            "{accessor-name_1}",
            "{accessor-name_1}"
        ],
        "r_accessor_permit": [
            {accessor-permit_1},
            {accessor-permit_2}
        ],
        "r_permit_type": [
            {permit-type_1},
            {permit-type_2}
        ]
        ...
    }
}
```



##### 2.4.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.4.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.4.1.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.4.1.4.3.8 Response body

When the response is successful, it contains all the attributes of the updated object.

##### Example 2-9: XML response

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<?xml version='1.0' encoding='UTF-8'?>
<acl xmlns="http://identifiers.emc.com/vocab/documentum"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="dm_acl" definition="{typeResourceUri}">
  <properties>
    <object_name>{object-name}</object_name>
    <description>{description of this object}</description>
    <owner_name>{object-owner}</owner_name>
    <r_accessor_name>
      <item>dm_world</item>
      <item>dm_owner</item>
      <item>{accessor-name_1}</item>
      <item>{accessor-name_2}</item>
    </r_accessor_name>
    <r_accessor_permit>
      <item>3</item>
      <item>7</item>
      <item>{accessor-permit_1}</item>
      <item>{accessor-permit_2}</item>
    </r_accessor_permit>
    <r_permit_type>
      <item>0</item>
      <item>0</item>
      <item>{permit-type_1}</item>
      <item>{permit-type_2}</item>
    </r_permit_type>
    ...
  </properties>
  <links>
    <link rel="self" href="{aclResourceUri}" />
    <link rel="edit" href="{aclResourceUri}" />
  </links>
</acl>
```

```

        <link rel="http://identifiers.emc.com/linkrel/delete"
              href="{aclResourceUri}"/>
        <link rel="http://identifiers.emc.com/linkrel/associations"
              href="{aclAssociationsResourceUri}"/>
    </links>
</acl>
```



#### ➡ Example 2-10: JSON response

```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
    "name": "acl",
    "type": "dm_acl",
    "definition": "{typeResourceUri}",
    "properties": {
        "object_name": "{object-name}",
        "description": "{description of this object}",
        "owner_name": "{object-owner}",
        "r_accessor_name": [
            "dm_world",
            "dm_owner",
            "{accessor-name_1}",
            "{accessor-name_2}"
        ],
        "r_accessor_permit": [
            3,
            7,
            {accessor-permit_1},
            {accessor-permit_2}
        ],
        "r_permit_type": [
            0,
            0,
            {permit-type_1},
            {permit-type_2}
        ],
        ...
    },
    "links": [
        {
            "rel": "self",
            "href": "{aclResourceUri}"
        },
        {
            "rel": "edit",
            "href": "{aclResourceUri}"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "{aclResourceUri}"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/associations",
            "href": "{aclAssociationsResourceUri}"
        }
    ]
}
```

```
    ]  
}
```



#### **2.4.1.4.4 Delete the ACL object instance**

Deletes the ACL object instance. When deletion fails, an exception is thrown.

##### **2.4.1.4.4.1 HTTP Request method and query parameters**

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	force	<p>Indicates whether or not to delete the ACL even when it is associated with other objects in the repository.</p> <p>After the deletion, objects associated with this ACL have no referenced ACL, which means that the following constraints apply to these objects:</p> <ul style="list-style-type: none"> <li>• These objects become read-only</li> <li>• These objects are accessible only to the object's owner and users with the SuperUser r privilege.</li> <li>• Only users with the SuperUser r privilege can delete these objects.</li> </ul>	string	<p><i>true</i>: Destroys the ACL object even if it is referenced by other objects in the repository. Only the super user has the necessary permissions to force delete an ACL object.</p> <p><i>false</i>: Does not destroy the ACL object if it is referenced by other objects.</p>	<i>false</i>

#### **2.4.1.4.4.2 Request media types**

N/A

#### **2.4.1.4.4.3 Request headers**

Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.4.1.4.4.4 Request body**

N/A

#### **2.4.1.4.4.5 Response headers**

N/A

#### **2.4.1.4.4.6 Response media types**

N/A

#### **2.4.1.4.4.7 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.4.1.4.4.8 Response body**

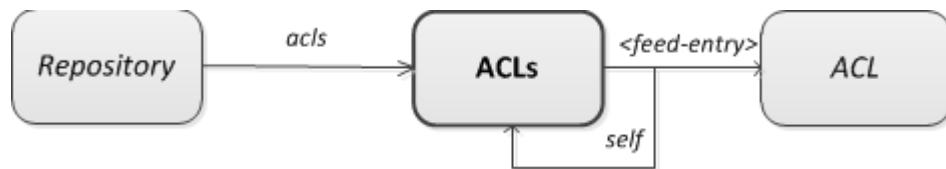
N/A

## **2.4.2 ACLs Collection**

The ACLs Collection represents the collection of all Access Control Lists (ACLs) in the repository that are available to the user that is currently logged into the system.

### 2.4.2.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.4.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: <Feed URI>
- Feed Title: ACLs
- Feed Updated: The server's current time
- Supports POST: Yes
- Entry: Each entry represents an ACL object
- Entry ID: The object ID of the ACL entry
- Entry Title: The object name of the ACL entry
- Entry Updated: The server's current time

### 2.4.2.3 Link relations

Link relation	Description	Resource reference
self	This ACL resource	“ACLs Collection” on page 50
first, last, next, previous	Pagination links	“ACLs Collection” on page 50

### 2.4.2.4 Operations

#### 2.4.2.4.1 Supported HTTP methods

Method	Description
POST	Creates an ACL
GET	Retrieves ACLs that are available to the current user from a repository

#### 2.4.2.4.2 Get the ACLs Collection

Lists all the available ACLs in the repository for the current user.

##### 2.4.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- sort
- page
- items-per-page
- include-total
- view
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.4.2.4.2.2 Request media types

N/A

##### 2.4.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.4.2.4.2.4 Request body**

N/A

**2.4.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.4.2.4.2.6 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.4.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.4.2.4.2.8 Response body**
 **Example 2-11: XML response**

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/acls</id>
  <title>Accls</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2015-06-09T10:20:41.905+00:00</updated>
  <dm:page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
  </dm:page>
  <dm:items-per-page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">2
  </dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO
    /acls?filter=owner_name=%27dave%27%20and%20r_is_internal=false
    &items-per-page=2"/>
  <link rel="next" href="http://localhost:8080/dctm-rest/repositories/REPO/
    acls?filter=owner_name=%27dave%27%20and%20r_is_internal=false
    &items-per-page=2"/>
```

```

        &items-per-page=2&page=2"/>
<link rel="first" href="http://localhost:8080/dctm-rest/repositories/REPO/
    acls?filter=owner_name=%27dave%27%20and%20r_is_internal=false
        &items-per-page=1"/>
<entry>
    <id>4500000580001d18</id>
    <title>test_acl_1377352452</title>
    <updated>2015-06-09T10:20:41.906+00:00</updated>
    <published>2015-06-09T10:20:41.906+00:00</published>
    <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/
acls/
        4500000580001d18"/>
    <content type="application/vnd.emc.documentum+xml"
        src="http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580001d18"/>
</entry>
<entry>
    <id>4500000580001d58</id>
    <title>3fb1de-8f77-4064-92a8-90d8950676</title>
    <updated>2015-06-09T10:20:41.906+00:00</updated>
    <published>2015-06-09T10:20:41.906+00:00</published>
    <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/
acls/
        4500000580001d58"/>
    <content type="application/vnd.emc.documentum+xml"
        src="http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580001d58"/>
</entry>
</feed>
```



### Example 2-12: JSON response

Status Code: 200 OK

Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/acls",
    "title": "ACLs",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2015-06-11T03:29:15.085+00:00",
    "page": 1,
    "items-per-page": 2,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls?
                filter=owner_name=%27dave%27%20and%20r_is_internal=false
                &items-per-page=2"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls?
                filter=owner_name=%27dave%27%20and%20r_is_internal=false
                &items-per-page=2&page=2"
        }
    ]
}
```

```
        },
        {
          "rel": "first",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/acls?
            filter=owner_name=%27dave%27%20and%20r_is_internal=false
            &items-per-page=2&page=1"
        }
      ],
      "entries": [
        {
          "id": "4500000580001d37",
          "title": "test_acl_1913536617",
          "updated": "2015-06-11T03:29:15.127+00:00",
          "published": "2015-06-11T03:29:15.127+00:00",
          "links": [
            {
              "rel": "edit",
              "href": "http://localhost:8080/dctm-rest/repositories/REPO/
acls/
4500000580001d37"
            }
          ],
          "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580001d37"
          }
        },
        {
          "id": "4500000580001d58",
          "title": "restAcl8f62ed43-f9",
          "updated": "2015-06-11T03:29:15.130+00:00",
          "published": "2015-06-11T03:29:15.130+00:00",
          "links": [
            {
              "rel": "edit",
              "href": "http://localhost:8080/dctm-rest/repositories/REPO/
acls/
4500000580001d58"
            }
          ],
          "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/acls/
4500000580001d58"
          }
        }
      ]
    }
  }
```



#### 2.4.2.4.3 Create an ACL

Create a new ACL in the repository.

##### 2.4.2.4.3.1 HTTP Request method and query parameters

POST

##### 2.4.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.4.2.4.3.3 Request headers

- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.4.2.4.3.4 Request body

- An XML or JSON message representing the ACL object that is created.
- The server throws an exception when undefined or read-only attributes are provided by the client.
- An ACL instance cannot be created manually. In other words, you cannot create an ACL with the value of the `acl_class` attribute set to 2. An ACL instance is automatically created when an ACL template is applied to a `SysObject` by the owner of the repository.
- The `r_object_type` attribute is not set because its value is `dm_acl` by default.

#### Example 2-13: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<acl xmlns="http://identifiers.emc.com/vocab/documentum"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="dm_acl"
      definition="{typeResourceUri}">
  <properties>
    <object_name>{object-name}</object_name>
    <description>{description of this object}</description>
    <owner_name>{object-owner}</owner_name>
    <r_accessor_name>
      <item>dm_world</item>
      <item>dm_owner</item>
      <item>{accessor-name_1}</item>
      <item>{accessor-name_2}</item>
    </r_accessor_name>
    <r_accessor_permit>
      <item>3</item>
      <item>7</item>
      <item>{accessor-permit_1}</item>
    </r_accessor_permit>
  </properties>
</acl>
```

```
<item>{accessor-permit_2}</item>
</r_accessor_permit>
<r_permit_type>
  <item>0</item>
  <item>0</item>
  <item>{permit-type_1}</item>
  <item>{permit-type_2}</item>
</r_permit_type>
...
</properties>
</acl>
```



#### ➤ Example 2-14: JSON request

```
{
  "name": "acl",
  "type": "dm_acl",
  "definition": "{typeResourceUri}",
  "properties": {
    "object_name": "{object-name}",
    "description": "{description of this object}",
    "owner_name": "{object-owner}",
    "r_accessor_name": [
      "dm_world",
      "dm_owner",
      "{accessor-name_1}",
      "{accessor-name_2}"
    ],
    "r_accessor_permit": [
      3,
      7,
      {accessor-permit_1},
      {accessor-permit_2}
    ],
    "r_permit_type": [
      0,
      0,
      {permit-type_1},
      {permit-type_2}
    ],
    ...
  }
}
```



#### 2.4.2.4.3.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.4.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.4.2.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.4.2.4.3.8 Response body

Every ACL has two built-in accessors:

- dm\_world
- dm\_owner

Even when these two accessors and their permissions are not provided by the client, the server grants them default permissions in the ACL object. The default permission value for each of these two accessors is defined by the configuration of the Documentum CM Server.

##### Example 2-15: XML response

Status Code: 201 Created

Location: `http://localhost/dctm-rest/repositories/REPO/acls/45000005800019c1`

Content-Type: `application/vnd.emc.documentum+json; charset=UTF-8`

```
<?xml version='1.0' encoding='UTF-8'?>
<acl xmlns="http://identifiers.emc.com/vocab/documentum"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="dm_acl"
      definition="{typeResourceUri}">
  <properties>
    <object_name>{object-name}</object_name>
```

```

<description>{description of this object}</description>
<owner_name>{object-owner}</owner_name>
<r_accessor_name>
<item>dm_world</item>
<item>dm_owner</item>
<item>{accessor-name_1}</item>
<item>{accessor-name_2}</item>
</r_accessor_name>
<r_accessor_permit>
<item>3</item>
<item>7</item>
<item>{accessor-permit_1}</item>
<item>{accessor-permit_2}</item>
</r_accessor_permit>
<r_permit_type>
<item>0</item>
<item>0</item>
<item>{permit-type_1}</item>
<item>{permit-type_2}</item>
</r_permit_type>
...
</properties>
<links>
<link rel="self" href="{aclResourceUri}" />
<link rel="edit" href="{aclResourceUri}" />
<link rel="http://identifiers.emc.com/linkrel/delete"
      href="{aclResourceUri}" />
<link rel="http://identifiers.emc.com/linkrel/associations"
      href="{aclAssociationsResourceUri}" />
</links>
</acl>

```



#### Example 2-16: JSON response

```

Status Code: 201 Created
Location: http://localhost/dctm-rest/repositories/REPO/acls/45000005800019c1
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "name": "acl",
  "type": "dm_acl",
  "definition": "{typeResourceUri}",
  "properties": {
    "object_name": "{object-name}",
    "description": "{description of this object}",
    "owner_name": "{object-owner}",
    "r_accessor_name": [
      "dm_world",
      "dm_owner",
      "{accessor-name_1}",
      "{accessor-name_2}"
    ],
    "r_accessor_permit": [
      3,
      7,
      {accessor-permit_1},
      {accessor-permit_2}
    ],
  }
}

```

```

    "r_permit_type": [
        0,
        0,
        {permit-type_1},
        {permit-type_2}
    ],
    ...
},
"links": [
    {
        "rel": "self",
        "href": "{aclResourceUri}"
    },
    {
        "rel": "edit",
        "href": "{aclResourceUri}"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "{aclResourceUri}"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/associations",
        "href": "{aclAssociationsResourceUri}"
    }
]
}

```



## 2.5 ACL Associations

The ACL Associations resource represents a collection of all the sysobjects that are associated with a specified ACL.

### 2.5.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[ACL](#)” on page 37 and [About the Diagram](#) on page 13.

## 2.5.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the ACL Associations resource
- Feed Title: Objects
- Feed Updated: The server's current time
- Supports POST: No
- Entry: “[SysObject](#)” on page 692
- Entry ID: The *<ObjectID>* of a SysObject
- Entry Title: The object name of SysObject
- Entry Updated: The value of the *<r\_modify\_date>*

## 2.5.3 Link relations

The following table lists general link relations in the ACL Associations resource:

Link relation	Description	Resource reference
self	The ACL Associations collection	“ <a href="#">ACL Associations</a> ” on page 60
first, last, next, previous	Pagination links	“ <a href="#">ACL Associations</a> ” on page 60

## 2.5.4 Operations

### 2.5.4.1 Supported HTTP methods

Method	Description
GET	GET all SysObjects associated with a specific ACL object.

### 2.5.4.2 List all available SysObjects associated with an ACL object

This method lists all available SysObjects that are associated with a specific ACL object.

#### **2.5.4.2.1 HTTP Request method and query parameters**

GET

This method supports the following common query parameters:

- inline
- sort
- page
- items-per-page
- include-total
- view
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### **2.5.4.2.2 Request media types**

N/A.

#### **2.5.4.2.3 Request headers**

- Accept
- Authorization

#### **2.5.4.2.4 Request body**

N/A.

#### **2.5.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.5.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.5.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.5.4.2.8 Response body

##### ➤ Example 2-17: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<feed
    xmlns="http://www.w3.org/2005/Atom"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>{aclAssociationsResourceUri}</id>
    <title>Objects</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>{current_time}</updated>
    <dm:page
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">{page}</dm:page>
    <dm:items-per-page
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">{items-per-page}</dm:items-per-page>
    <link rel="self" href="{aclAssociationsResourceUri}" />
    ... pagination links ...
    <entry>
        <id>{objectResourceUri}</id>
        <title>{object_name}</title>
        <author>
            <name>{author_name}</name>
            <uri>{userResourceUri}</uri>
        </author>
        <summary>{r_object_type} {r_object_id}</summary>
        <updated>{r_modify_date} or {current_time}</updated>
        <published>{creation-date} or {current_time}</published>
        <link rel="edit" href="{objectResourceUri}" />
        <content type="application/vnd.emc.documentum+xml"
            src="{objectResourceUri}" />
    </entry>
    ...
</feed>
```



 **Example 2-18: JSON response**

```
{  
    "id": "{aclAssociationsResourceUri}",  
    "title": "Objects",  
    "author": [  
        {  
            "name": "OpenText Documentum"  
        }  
    ],  
    "updated": "{current_time}",  
    "page": {page},  
    "items-per-page": {items-per-page},  
    "links": [  
        {  
            "rel": "self",  
            "href": "{aclAssociationsResourceUri}"  
        },  
        ... pagination links ...  
    ],  
    "entries": [  
        {  
            "id": "{objectResourceUri}",  
            "title": "{object_name}",  
            "author": [  
                {  
                    "name": "{author_name}",  
                    "uri": "{userResourceUri}"  
                }  
            ],  
            "summary": "{r_object_type} {r_object_id}",  
            "updated": "{r_modify_date} or {current_time}",  
            "published": "{creation-date} or {current_time}",  
            "links": [  
                {  
                    "rel": "edit",  
                    "href": "{objectResourceUri}"  
                }  
            ],  
            "content": {  
                "type": "application/vnd.emc.documentum+json",  
                "src": "{objectResourceUri}"  
            }  
        },  
        ...  
    ]  
}
```



## 2.6 Archived Contents

This resource uses a DEFLATE compression algorithm to download the contents of multiple objects, and returns that content to the client as a ZIP file formatted stream. This resource can download the content media of all content from the REST server using one request instead of an OpenText™ Documentum™ Content Management Accelerated Content Services server or a OpenText™ Documentum™ Content Management Branch Office Caching Services server.

This resource downloads the CURRENT version of an object. When a folder, or a subtype of *<dm\_folder>*, has been specified, the request to the REST server downloads the contents of the objects that are directly under the specified folder. However, the object contents of sub folder are downloaded.

### 2.6.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and [About the Diagram](#) on page 13.

### 2.6.2 Feed

Is this resource a feed? Yes.

### 2.6.3 Link relations

N/A

### 2.6.4 Operations

#### 2.6.4.1 Supported HTTP methods

Method	Description
GET	Using a ZIP file format, this method retrieves all of the contents from a single object.
POST	Using a ZIP file format, this method retrieves the first page of the contents, or all of the contents, from multiple objects.

#### 2.6.4.1.1 rest.archive.contents.threshold

In order to prevent potential performance issues caused by attempting to download huge files, the REST server uses a runtime property called `rest.archive.contents.threshold` to set the maximum file size allowed for content. The maximum allowed file size defaults to 20 GB and is set using `<kilobytes>`. Exceeding the maximum file size causes an error to be returned by the REST server, and archiving fails.

##### Example 2-19: rest-api-runtime.properties

```
# Sets the maximum size, in megabytes, of the contents that are allowed to be
# archived.
# When the sum of all the contents' original size is greater than the value of
# ${rest.archive.contents.threshold} megabytes, then that contents will not be
# archived
# and an error is returned by the REST server.
# The value of this runtime property defaults to 20971520 kilobytes, which is
# 20GB.
#
rest.archive.contents.threshold=20971520
```



#### 2.6.4.2 Get a single object's contents

Retrieves all of the contents of a single object. The specified object can be a simple document, a folder, or a virtual document. When the object is a folder, or a subtype of `<dm_folder>`, then the REST server retrieves all of the objects contents (`<dm_sysobject>`) that are directly under that folder, in this case, any virtual documents that exist under this folder are treated as simple documents. When the specified object is a virtual document, you can use the `<include-children>` parameter to determine whether or not contents that are in its child objects is also downloaded.

##### 2.6.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	object-id	Defines the object ID of the object whose contents the client wants to get.	string	N/A	null
	format	Specifies the file format of the contents that is retrieved.	string	Any format that is registered in the Format table of the repository.	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	include-children	When the object is a virtual document this parameter determines whether or not to get the content from its children.	string	<i>true, false</i>	<i>false</i>
	include-broken	Determines whether or not to include nodes with broken bindings for a virtual document.	string	<i>true, false</i>	<i>false</i>
	binding-label	Determines the version label that is used to resolve late bound nodes for a virtual document.	string		CURRENT
	depth	Defines to what depth level in the virtual document hierarchy will be returned.  A value of -1 is used to return the whole virtual document tree.	integer	-1 or $\geq 0$	1



**Note:** The `<include-broken>`, `<binding-label>`, and `<depth>` parameters only work when the object is a virtual document and the `<include-children>` parameter has a value of *true*.

#### **2.6.4.2.2 Request media types**

N/A

#### **2.6.4.2.3 Request headers**

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.6.4.2.4 Request body**

N/A

#### **2.6.4.2.5 Response headers**

- Content-Type: application/zip
- Content-Disposition: attachment;filename={object-name}\_content.zip

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.6.4.2.6 Response media types**

N/A

#### **2.6.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.6.4.2.8 Response body**

##### **2.6.4.2.8.1 Naming conventions**

Each item in a ZIP file must have a unique name. To ensure that each content object item in the ZIP file has a unique name, the REST server follows these naming conventions to generate each content object name:

- By default, each content object is named using the following pattern: {object-name}.{extension}
- Here's an example to illustrate the naming conventions.
  - Let's say that there is a folder named <mmtm\_deck> in the home cabinet, and the folder id is <0b0000058000951d>.

- The above folder has 7 documents in it. Some of those documents have a file name and extension and others do not. The following image shows this folder:

The screenshot shows the DA interface with a tree view on the left and a grid view on the right. The tree view shows a path: Saved Searches, Subscriptions, My Files, Categories, Administration, My Home Cabinet, Cabinets, Administrator, demo, Demo, mmtm\_deck. A red box highlights the 'mmtm\_deck' folder. The grid view is titled 'Folder structure in DA' and lists the following files:

Name	Full Content_size	Format
CityofAtlanta.pptx ①	2.45 MB	MS PowerPoint Presentation 2007 / 2010
NetTopology ②*	73 KB	Portable Network Graphics
NetTopology ③*	1.54 MB	JPEG Image
NetTopology ④*	1.39 MB	JPEG Image
NuclearSafetyAuthority.ppt ⑤	7.78 MB	MS PowerPoint pre-3.0 (MacOS, Windows)
NuclearSafetyAuthority_attachment ⑥*	248 KB	Portable Network Graphics
XLGroup ⑦*	5.91 MB	MS PowerPoint Presentation 2007 / 2010

In the above image, we have numbered the files so we can see the resulting file names in the next image. Take particular note of files number 3 and 4 above, they both have the same file name and file extension.

- At this point, we use the REST API to download the above folder's contents to our local file system using the following GET method statement:

```
/dcm-rest/repositories/REPO/archived-contents?object-id=0b0000058000951d
```

- The response is a ZIP file that we can save to our local file system. The suggested name for the resulting ZIP file is `mmtm_deck_content.zip`.
- When we extract the ZIP file and examine the folder contents, we see the folder structure shown in the image below. Note the resulting names for files that would have caused a name duplication, such as file number 4 (NetTopology), which has had its name generated using its object ID.

The screenshot shows a Windows File Explorer window with the title 'mmtm\_deck\_content.zip'. The left sidebar shows 'Favorites' (Desktop, Downloads, Recent Places), 'Libraries' (Documents, Music, Pictures, Videos), and 'Computer'. The main area shows a list of files from the ZIP archive:

Name	Type	Size	Date modified
CityofAtlanta.pptx ①	PPTX File	2,518 KB	10/11/2017 10:30 AM
NetTopology(06000005800076c5).jpg ④*	JPEG image	1,429 KB	10/11/2017 10:30 AM
NetTopology.jpg ③*	JPEG image	1,579 KB	10/11/2017 10:30 AM
NetTopology.png ②*	PNG image	73 KB	10/11/2017 10:30 AM
NuclearSafetyAuthority.ppt ⑤	PPT File	7,976 KB	10/11/2017 10:30 AM
NuclearSafetyAuthority_attachment.png ⑥*	PNG image	248 KB	10/11/2017 10:30 AM
XLGroup.pptx ⑦*	PPTX File	6,062 KB	10/11/2017 10:30 AM

A red box highlights the 'Extracted folder contents in client OS' text at the bottom of the window.

#### 2.6.4.2.8.2 Special cases

- When an object name is null or empty, the string “<\_EMPTY\_>” is used to replace the variable <{object-name}>, which results in the name <\_EMPTY\_.{extension}>.
- Since ZIP uses slashes as reserved characters to mark directories, all forward slash (“\”) characters in object names are replaced with their Unicode equivalent of “%2F”, and all back slash (“\\”) characters are replaced with “%5C”.
- When the content format parameter is empty or null, the “<.{extension}>” portion of the object name is not used or displayed.
- When a duplication of any ZIP entry with <{object-name}.{extension}> occurs, the content object id is appended to the name to ensure that it is unique. For example, a duplicated <{object-name}.{extension}> entry name would be changed into <{object-name}({content-id}).{extension}> to make it unique.

#### 2.6.4.2.9 Code samples

##### ► Example 2-20: Request and response

```
// Request
GET /dctm-rest/repositories/REPO/archived-contents?object-id=0b0004d2800060e6
HTTP/1.1
Host: localhost:8080
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==

// Response
Status 200 OK

content-disposition →attachment; filename="test-folder-1_content.zip"
content-type →application/zip
```

... zip file format stream ...



#### 2.6.4.3 Get multiple object's contents

##### 2.6.4.3.1 HTTP Request method and query parameters

Retrieves the first page of contents or all of the contents for multiple objects in ZIP file format. When an object is a folder (or a subtype of <dm\_folder>), the REST server gets the contents of those objects (<dm\_sysobject>) that are under that folder. When this is the case, virtual documents are treated as simple documents.

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	format	Specifies the file format of the contents that is retrieved.	string	Any format that is registered in the Format table of the repository	null
	all-contents	Determines whether to retrieve all of the object's contents, or only retrieve the first page contents.	string	<i>true, false</i>	<i>false</i>

#### 2.6.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.6.4.3.3 Request headers

- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.6.4.3.4 Request body

The Request body contains a list of URLs for the object, document, or folder that the client wants to download. The contents of the object is downloaded only once, even when a duplicated object id is provided by the client.

##### ➡ Example 2-21: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
  <hrefs>
    <href>http://localhost:8080/dctm-rest/repositories/REPO/objects/
      090004d280005520</href>
    <href>http://localhost:8080/dctm-rest/repositories/REPO/documents/
      090004d280005521</href>
    <href>http://localhost:8080/dctm-rest/repositories/REPO/folders/
      0b0004d2800060e8</href>
  </hrefs>
</content>
```



 **Example 2-22: JSON request**

```
{  
    "hrefs":  
    [  
        "http://localhost:8080/dctm-rest/repositories/REPO/objects/  
        090004d280005520",  
        "http://localhost:8080/dctm-rest/repositories/REPO/documents/  
        090004d280005521",  
        "http://localhost:8080/dctm-rest/repositories/REPO/folders/  
        0b0004d2800060e8"  
    ]  
}
```

**2.6.4.3.5 Response headers**

- Content-Type: application/zip
- Content-Disposition: attachment;filename=multiple\_objects\_contents.zip

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.6.4.3.6 Response media types**

N/A

**2.6.4.3.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.6.4.3.8 Response body**

The Response returns a formatted stream of the object contents as a ZIP file. When the specified object is a folder, then all of the contents of the specified object are located within a folder with the same name as the object in the ZIP file.

When an object name duplication exists within the ZIP file, the folder's name is generated using the object-id. For example, a folder called test-folder, whose id is 0b0004d2800060e7 would be named test-folder(0b0004d2800060e7) . Each file name within the ZIP file must be unique and must also comply with the naming conventions listed in ["Get a single object's contents" on page 66](#).

#### 2.6.4.3.9 More samples

➡ Example 2-23: Request and response

```
// Request
POST /dctm-rest/repositories/REPO/archived-contents?all-contents=true HTTP/1.1
Host: localhost:8080
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
Content-Type: application/vnd.emc.documentum+json

{
    "hrefs":
    [
        "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090004d280005520",
        "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090004d280005521",
        "http://localhost:8080/dctm-rest/repositories/REPO/documents/
090004d2800055d"
    ]
}

// Response
Status 200 OK

content-disposition attachment; filename="multiple_object_contents.zip"
content-type application/zip
```

... zip file format stream ...



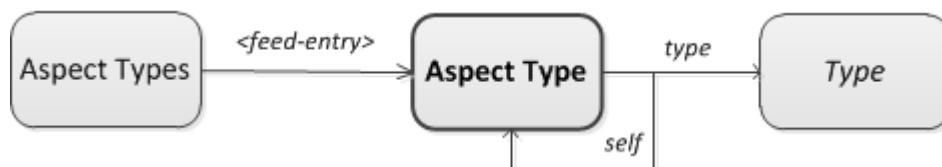
## 2.7 Aspect Type (s)

### 2.7.1 Aspect Type

The Aspect Type resource represents an aspect type in Documentum CM Server.

#### 2.7.1.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also: “Aspect Types” on page 78 and About the Diagram on page 13.*

### 2.7.1.2 Feed

No feed support.

### 2.7.1.3 Link relations

The following table lists general link relations in the Aspect Type resource.

Link relation	Description	Resource reference
self	This aspect type.	<a href="#">"Aspect Type" on page 73</a>
type [1]	When an aspect type defines its own properties, this link relation points to those properties.	<a href="#">"Type" on page 727</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
`http://identifiers.emc.com/linkrel/`.

### 2.7.1.4 Operations

#### 2.7.1.4.1 Supported HTTP methods

Method	Description
GET	GET an aspect type.

#### 2.7.1.4.2 Get an aspect type

GET the metadata of an Aspect Type resource.

##### 2.7.1.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
GET	locale	Specifies the value of a valid locale for the Request. For example, it can be used to get the label text of an aspect type object.	string		null
	inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	view				

#### 2.7.1.4.2.2 Request media types

N/A

#### 2.7.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.7.1.4.2.4 Request body

N/A

#### 2.7.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.7.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.7.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.7.1.4.2.8 Response body

XML or JSON representation of the Aspect Type resource:

- When the aspect type defines additional properties, the type link relation points to the Type resource that contains the properties that this Aspect Type resource defines.
- The properties block contains the following two sets of data:
  - The metadata of this Aspect Type resource.
  - The properties that this Aspect Type resource defines. These properties are not presented one by one. Instead, an Aspect Type resource displays the

metadata of these properties in eight elements, all of which are prefixed with attr\_ as shown here:

- Name of the property as shown in the attr\_name element
- Whether the property is a repeating property as shown in the attr\_repeating element
- Data type of the properties as shown in the attr\_type element
- Length of the properties as shown in the attr\_length element
- Label of the property as shown in the attr\_label element
- Whether the property is a hidden property as shown in the attr\_hidden element
- Whether the property is a non-null property as shown in the attr\_notnull element
- Whether the property is a read-only property as shown in the attr\_READONLY element

These eight elements contain the same number of items, which is also the number of properties this aspect type defines. The combination of items in the same position of these elements represents one property this aspect type defines. In the following sample, the combination of all first items indicates that this resource defines a string property named annotation\_name. The property annotation\_name is a repeating property with a maximum length of 225.



**Note:** When the value of an element that is prefixed with attr\_is null, that element is not shown in the Response. That is why the following code sample only contains four elements.

#### XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<aspect-type
    definitions="http://localhost:8080/dctm-rest/repositories/REPO/types/
dmc_aspect_type.xml"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="dmc_aspect_type">
<properties>
    <group_name>docu</group_name>
    <acl_name>BOF_acl</acl_name>
    <world_permit>3</world_permit>
    <creation-date>2015-03-03T02:29:34.000+00:00</creation-date>
    <r_object_type>dmc_aspect_type</r_object_type>
    <min_dfc_version>6.0</min_dfc_version>
    <owner_name>Administrator</owner_name>
    <implementation_technology>Java</implementation_technology>
    <primary_class>com.documentum.smart.impl.aspect.AnnotatableAspect</primary_class>
    <owner_permit>7</owner_permit>
    <a_bof_version/>
    <a_interfaces xsi:nil="true"/>
    <r_object_id>0b000005800005b0</r_object_id>
    <r_modify_date>2015-03-03T02:30:13.000+00:00</r_modify_date>
    <group_permit>1</group_permit>
    <i_folder_id>
        <item>0b00000580000180</item>
```

```

</i_folder_id>
<r_folder_path>
    <item>/System/Modules/Aspect/my_aspect_1</item>
</r_folder_path>
<object_name>my_aspect_1</object_name>
<target_object_type xsi:nil="true"/>
<r_creator_name>Administrator</r_creator_name>
<r_modifier>Administrator</r_modifier>
<acl_domain>REPO_ADMIN</acl_domain>
<a_module_type>Aspect</a_module_type>
<attr_name>
    <item>annotation_name</item>
    <item>module_name</item>
    <item>annotation_metadata</item>
    <item>i_partition</item>
</attr_name>
<attr_repeating>
    <item>true</item>
    <item>true</item>
    <item>true</item>
    <item>false</item>
</attr_repeating>
<attr_type>
    <item>string</item>
    <item>string</item>
    <item>string</item>
    <item>integer</item>
</attr_type>
<attr_length>
    <item>255</item>
    <item>255</item>
    <item>1024</item>
    <item xsi:nil="true"/>
</attr_length>
</properties>
<links>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/aspect-types/
my_aspect_1"
        rel="self"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/types/
dmi_03000005800001d3"
        rel="http://identifiers.emc.com/linkrel/type"/>
</links>
</aspect-type>

```

*JSON response*

```
{
    "name" : "aspect-type",
    "type" : "dmc_aspect_type",
    "definition" : "http://localhost:8080/dctm-rest/repositories/REPO/types/
dmc_aspect_type",
    "properties" : {
        "group_name" : "docu",
        "acl_name" : "BOF_acl",
        "world_permit" : 3,
        "creation-date" : "2015-03-03T02:29:34.000+00:00",
        "r_object_type" : "dmc_aspect_type",
        "min_dfc_version" : "6.0",
        "owner_name" : "Administrator",
        "implementation_technology" : "Java",
        "primary_class" : "com.documentum.smart.impl.aspect.AnnotatableAspect",
        "owner_permit" : 7,
        "a_bof_version" : "",
        "a_interfaces" : null,
        "r_object_id" : "0b000005800005b0",
        "r_modify_date" : "2015-03-03T02:30:13.000+00:00",
        "group_permit" : 1,
    }
}
```

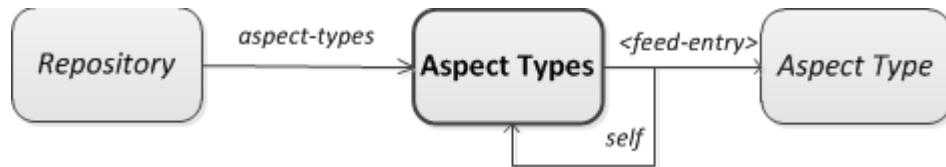
```
"i_folder_id" : [
    "0b00000580000180"
],
"r_folder_path" : [
    "/System/Modules/Aspect/my_aspect_1"
],
"object_name" : "my_aspect_1",
"target_object_type" : null,
"r_creator_name" : "Administrator",
"r_modifier" : "Administrator",
"acl_domain" : "REPO_ADMIN",
"a_module_type" : "Aspect",
"attr_name" : [
    "annotation_name",
    "module_name",
    "annotation_metadata",
    "i_partition"
],
"attr_repeating" : [
    true,
    true,
    true,
    false
],
"attr_type" : [
    "string",
    "string",
    "string",
    "integer"
],
"attr_length" : [
    255,
    255,
    1024,
    null
]
},
"links" : [
{
    "rel" : "self",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/aspect-types/
        my_aspect_1"
},
{
    "rel" : "http://identifiers.emc.com/linkrel/type",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dmi_03000005800001d3"
}
]
```

## 2.7.2 Aspect Types

The Aspect Types resource represents a collection of aspect type resources.

### 2.7.2.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.7.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the Aspect Types resource without the file extension
- Feed Title: Aspect types
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Aspect Type” on page 73
- Entry ID: URI of the Aspect Type
- Entry Title: Aspect Type name
- Entry Updated: r\_modify\_date of the Aspect Type

### 2.7.2.3 Link relations

The following table lists general link relations in the Aspect Types resource.

Link relation	Description	Resource reference
self	This Aspect Types resource	“Aspect Types” on page 78
first, last, next, previous	Pagination links	“Aspect Types” on page 78

### 2.7.2.4 Operations

#### 2.7.2.4.1 Supported HTTP methods

Method	Description
GET	GET all aspect types in a repository

#### 2.7.2.4.2 Get aspect types

GET all aspect types in a repository.

##### 2.7.2.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters
GET	inline sort page items-per-page include-total view filter links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

##### 2.7.2.4.2.2 Request media types

N/A

##### 2.7.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.7.2.4.2.4 Request body

N/A

#### 2.7.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.7.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing)

#### 2.7.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.7.2.4.2.8 Response body

XML or JSON representation of the Aspect Types collection.

Properties that an Aspect Type resource defines, which are prefixed with attr\_ are not displayed in the response. Additionally, these properties do not take effect in the view, filter, and sort parameters.

## 2.8 Audit Policy (ies)

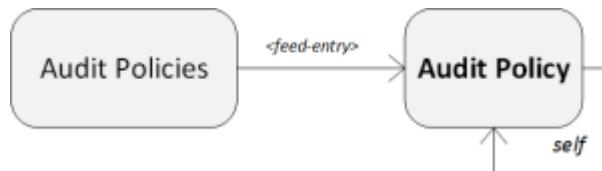
### 2.8.1 Audit Policy

The Audit Policy resource displays the deletion policy that applies to audit records for a specific user or group. Only an authorized user or group can delete an audit record. When there are multiple policies that apply to the same user, the policy with the greatest permission priority is in effect. All users can view audit policies, but only a user who is in the Install Owner role can access, create, update, or delete an audit policy.

If an unauthorized user or group attempts to delete an audit record, the Documentum CM Server throws an exception and error message.

### 2.8.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Audit Policies” on page 88 and About the Diagram on page 13.

### 2.8.1.2 Feed

No feed support.

### 2.8.1.3 Link relations

Link relation	Description	Resource reference
self	This resource	“Audit Policy” on page 81
delete [1]	Delete this Audit Policy object	“Audit Policy” on page 81
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code>		

### 2.8.1.4 Operations

#### 2.8.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a single Audit Policy object using its ID.
POST	Update the current Audit Policy for specified attributes.
DELETE	Delete the current Audit Policy object.

#### 2.8.1.4.2 Get an audit policy

This operation retrieves a specified audit policy object.

##### 2.8.1.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
GET	links	<p>For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a>.</p> <p> <b>Note:</b> The view parameter is invalid because this resource displays information in a customized manner.</p>			

##### 2.8.1.4.2.2 Request media types

N/A

##### 2.8.1.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.8.1.4.2.4 Request body

N/A

##### 2.8.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.8.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.8.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.8.1.4.2.8 Response body

##### Example 2-24: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-policy xmlns="http://identifiers.emc.com/vocab/documentum"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <object-id>0002791780000d00</object-id>
  <name>AuditAdminCanPurgeDocument</name>
  <accessor-name>auditadmin</accessor-name>
  <is-group>false</is-group>
  <attribute-rules>
    <time_stamp_utc>2017-06-20 16:00:00</time_stamp_utc>
    <object_type>dm_user</object_type>
  </attribute-rules>
  <links>
    <link rel="self"
      href="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
      0002791780000d00" />
  </links>
</audit-policy>
```



##### Example 2-25: JSON

```
{
  "object-id": "0002791780000d00",
  "name": "AuditAdminCanPurgeDocument",
  "accessor-name": "auditadmin",
  "is-group": false,
  "attribute-rules": {
    "time_stamp_utc": "2017-06-20 16:00:00",
    "object_type": "dm_user"
  },
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
      0002791780000d00"
    }
  ]
}
```



#### 2.8.1.4.3 Update an audit policy

This operation updates a specified Audit Policy object.

##### 2.8.1.4.3.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
POST	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.8.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.8.1.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.8.1.4.3.4 Request body

###### Example 2-26: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-policy xmlns="http://identifiers.emc.com/vocab/documentum"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="dm_audit_policy"
               definition="http://localhost:8080/dctm-rest/repositories/repos/
types/
                           dm_audit_policy">
    <name>testPolicy</name>
    <accessor_name>auditadmin</accessor_name>
    <attribute-rules>
        <time_stamp_utc>2017-06-20 16:00:00</time_stamp_utc>
        <object_type>dm_user</object_type>
    </attribute-rules>
</audit-policy>
```



###### Example 2-27: JSON

```
{
    "name": "AuditAdminCanPurgeDocument",
    "accessor-name": "auditadmin",
    "attribute-rules": {
        "time_stamp_utc": "2017-06-20 16:00:00",
        "object_type": "dm_user"
```

```
}
```



#### 2.8.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.8.1.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.8.1.4.3.8 Response body

##### ► Example 2-28: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-policy xmlns="http://identifiers.emc.com/vocab/documentum"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <object-id>0002791780000d00</object-id>
    <name>AuditAdminCanPurgeDocument</name>
    <accessor-name>auditadmin</accessor-name>
    <is-group>false</is-group>
    <attribute-rules>
        <time_stamp_utc>2017-06-20 16:00:00</time_stamp_utc>
        <object_type>dm_user</object_type>
    </attribute-rules>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
              0002791780000d00" />
    </links>
  </audit-policy>
```



 **Example 2-29: JSON**

```
{
  "object-id": "0002791780000d00",
  "name": "AuditAdminCanPurgeDocument",
  "accessor-name": "auditadmin",
  "is-group": false,
  "attribute-rules": [
    {
      "time_stamp_utc": "2017-06-20 16:00:00",
      "object_type": "dm_user"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
          0002791780000d00"
    }
  ]
}
```



#### 2.8.1.4.4 Delete an Audit Policy

This operation deletes a specified Audit Policy object.

##### 2.8.1.4.4.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
DELETE	N/A	<p>For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a>.</p> <p> <b>Note:</b> The view parameter is invalid because this resource displays information in a customized manner.</p>			

##### 2.8.1.4.4.2 Request media types

N/A

##### 2.8.1.4.4.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.1.4.4.4 Request body

N/A

#### 2.8.1.4.4.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.1.4.4.6 Response media types

N/A

#### 2.8.1.4.4.7 Response status

- 204 - Delete successful
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.8.1.4.4.8 Response body

N/A

## 2.8.2 Audit Policies

The Audit Policies resource shows all audit policies that apply to the repository. Only a user with the Install Owner role can access, create, update, or delete an audit policy. All other users may only view the audit policy.

### 2.8.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and [About the Diagram on page 13](#).

### 2.8.2.2 Feed

Is this resource a feed? Yes

- Feed ID: Feed URI
- Feed Title: Audit Policies
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: Audit Policy
- Entry ID: URI of the audit policy
- Entry Title: The ID of a specified audit policy
- Entry Updated: Current server time

### 2.8.2.3 Link relations

Link relation	Description	Resource reference
self	This resource	"Audit Policies" on page 88

### 2.8.2.4 Operations

#### 2.8.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves all Audit Policies from the repository.
POST	Create a new Audit Policy.

#### 2.8.2.4.2 Get all audit policies

This operation retrieves all Audit Policies from the repository.

##### 2.8.2.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• inline</li> <li>• page</li> <li>• items-per-page</li> <li>• include-total</li> <li>• filter</li> <li>• links</li> </ul>	<p>For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.</p> <p> <b>Note:</b> The view parameter is not supported because this resource displays information in a customized manner.</p>			

#### 2.8.2.4.2.2 Request media types

N/A

#### 2.8.2.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.2.4.2.4 Request body

N/A

#### 2.8.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.2.4.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.8.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.8.2.4.2.8 Response body

##### ➡ Example 2-30: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/repo/audit-policies</id>
  <title>Audit Policies</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2017-06-05T09:29:00.096+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
```

```

        100
    </dm:items-per-page>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies"/>
    <entry>
        <id>
            http://localhost:8080/dctm-rest/repositories/repo/audit-policies/
0002791780000d17
        </id>
        <title>0002791780000d17</title>
        <summary>Audit Purge Policy</summary>
        <link rel="edit"
              href="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
            0002791780000d17" />
        <content type="application/vnd.emc.documentum+xml"
src="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
            0002791780000d17" />
    </entry>
    <entry>
        <id>
            http://localhost:8080/dctm-rest/repositories/repo/audit-policies/
0002791780000d19
        </id>
        <title>0002791780000d19</title>
        <summary>Audit Purge Policy</summary>
        <link rel="edit"
              href="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
            0002791780000d19" />
        <content type="application/vnd.emc.documentum+xml"
src="http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
            0002791780000d19" />
    </entry>
</feed>
```



### Example 2-31: JSON

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/repo/audit-policies",
    "title": "Audit Policies",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2017-06-05T09:28:33.094+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies"
        }
    ],
}
```

```
"entries": [
  {
    "id": "http://localhost:8080/dctm-rest/repositories/repo/audit-policies/
          0002791780000d17",
    "title": "0002791780000d17",
    "summary": "Audit Purge Policy",
    "links": [
      {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
          0002791780000d17"
      }
    ],
    "content": {
      "type": "application/vnd.emc.documentum+json",
      "src": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
          0002791780000d17"
    }
  },
  {
    "id": "http://localhost:8080/dctm-rest/repositories/repo/audit-policies/
          0002791780000d19",
    "title": "0002791780000d19",
    "summary": "Audit Purge Policy",
    "links": [
      {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
          0002791780000d19"
      }
    ],
    "content": {
      "type": "application/vnd.emc.documentum+json",
      "src": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
          0002791780000d19"
    }
  }
]
```



#### 2.8.2.4.3 Create an audit policy

This operation creates a specified Audit Policy object.

#### 2.8.2.4.3.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
POST	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.8.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.8.2.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.2.4.3.4 Request body

##### Example 2-32: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-policy xmlns="http://identifiers.emc.com/vocab/documentum"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="dm_audit_policy"
               definition="http://localhost:8080/dctm-rest/repositories/repos/
types/                               dm_audit_policy">
    <name>testPolicy</name>
    <accessor-name>auditadmin</accessor-name>
    <attribute-rules>
        <time_stamp_utc>2017-06-20 16:00:00</time_stamp_utc>
        <object_type>dm_user</object_type>
    </attribute-rules>
</audit-policy>
```



##### Example 2-33: JSON

```
{
    "name": "AuditAdminCanPurgeDocument",
    "accessor-name": "auditadmin",
    "attribute-rules": {
        "time_stamp_utc": "2017-06-20 16:00:00",
        "object_type": "dm_user"
    }
}
```



#### 2.8.2.4.3.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.8.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.8.2.4.3.7 Response status

- 201 - Create successful
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.8.2.4.3.8 Response body

##### Example 2-34: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-policy xmlns="http://identifiers.emc.com/vocab/documentum"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <object-id>0002791780000d00</object-id>
  <name>AuditAdminCanPurgeDocument</name>
  <accessor-name>auditadmin</accessor-name>
  <is-group>false</is-group>
  <attribute-rules>
    <time_stamp_utc>2017-06-20 16:00:00</time_stamp_utc>
    <object_type>dm_user</object_type>
  </attribute-rules>
  <links>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/repo/audit-
polices/
          0002791780000d00" />
  </links>
</audit-policy>
```



##### Example 2-35: JSON

```
{
  "object-id": "0002791780000d00",
  "name": "AuditAdminCanPurgeDocument",
  "accessor-name": "auditadmin",
```

```

    "is-group": false,
    "attribute-rules": {
        "time_stamp_utc": "2017-06-20 16:00:00",
        "object_type": "dm_user"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-
policies/
0002791780000d00"
        }
    ]
}

```



## 2.9 Audit Trail (s)

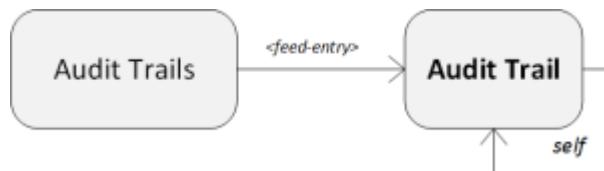
### 2.9.1 Audit Trail

The Audit Trail resource is a record of user operations that occurred within the repository of “[Registered Audit Events](#)” on page 524 and certain “[Audit Policies](#)” on page 88 resources.

Logged in users must have VIEW privileges in their extended user privileges to access to this resource. In addition, logged in users must have PURGE privileges to gain access to the DELETE link for certain Audit Trail objects. Otherwise, the DELETE link is not visible in those Audit Trail objects.

#### 2.9.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Audit Trails](#)” on page 102 and [About the Diagram](#) on page 13.

### 2.9.1.2 Feed

No feed support.

### 2.9.1.3 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">"Audit Trail" on page 95</a>
delete [1]	Delete this audit trail	N/A
author	The author of this audit trail	<a href="#">"User" on page 766</a>
audited-object [1]	The audited object of this audit trail	<a href="#">"SysObject" on page 692</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
`http://identifiers.emc.com/linkrel/.`

### 2.9.1.4 Operations

#### 2.9.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the audit trail object.
DELETE	Purges the audit trail object.

#### 2.9.1.4.2 Get an audit trail

##### 2.9.1.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.9.1.4.2.2 Request media types

N/A

#### 2.9.1.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.9.1.4.2.4 Request body

N/A

#### 2.9.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.9.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.9.1.4.2.7 Response status

- 200 - Retrieved successfully.
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.9.1.4.2.8 Response body

##### ➡ Example 2-36: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-trail xmlns="http://identifiers.emc.com/vocab/documentum"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:type="dm_audittrail"
              definition="http://localhost:8080/dctm-rest/repositories/REPO/
types/
                                         dm_audittrail"
<properties>
    <event_name>dm_save</event_name>
    <event_source>System Unspecific</event_source>
    <r_gen_source>1</r_gen_source>
    <user_name>Administrator</user_name>
    <audited_obj_id>0900000580000ce9</audited_obj_id>
    <time_stamp>2015-03-03T02:34:20.000+00:00</time_stamp>
    <string_1>Create</string_1>
    <id_1>00000000000000000000000000000000</id_1>
```

```
<id_2>0000000000000000</id_2>
<id_3>0000000000000000</id_3>
<id_4>0000000000000000</id_4>
<id_5>0000000000000000</id_5>
<chronicle_id>0900000580000ce9</chronicle_id>
<object_name>Result.dm_FTQBS_HOURLY</object_name>
<version_label>1.0</version_label>
<object_type>dm_document</object_type>
<event_description>Default Event Set</event_description>
<policy_id>0000000000000000</policy_id>
<workflow_id>0000000000000000</workflow_id>
<session_id>010000058000015b</session_id>
<user_id>1100000580000102</user_id>
<owner_name>Administrators</owner_name>
<acl_name>dm_4500000580000101</acl_name>
<acl_domain>Administrator</acl_domain>
<attribute_list>
    acl_name="dm_4500000580000101",
    acl_domain="Administrator",
    world_permit=3,
    group_permit=5,
    group_name="docu",
    owner_permit=7,
    owner_name="Administrator",
    a_storage_type="filestore_01",
    a_full_text=0,
    a_content_type="crtext",
    subject="Result of dm_method(dm_FTQBS_HOURLY) with command line:
        -method_verb
com.documentum.server.impl.method.fulltext.qbs.FTQBSJob
        -__dm_docbase__ REPO -__dm_server_config__ REPO -docbase_name
R....",
    title="Result of dm_method(dm_FTQBS_HOURLY) with status code
(0)",
    object_name="Result.dm_FTQBS_HOURLY"
</attribute_list>
<attribute_list_id>0000000000000000</attribute_list_id>
<audit_version>4</audit_version>
<host_name>restcs72ga</host_name>
<time_stamp_utc>2015-03-03T02:34:20.000+00:00</time_stamp_utc>
<i_audited_obj_class>0</i_audited_obj_class>
<registry_id>2600000580000101</registry_id>
<i_is_archived>false</i_is_archived>
<audited_obj_vstamp>0</audited_obj_vstamp>
<attribute_list_aspect_id>0000000000000000</attribute_list_aspect_id>
<r_object_sequence>4</r_object_sequence>
<i_is_replica>false</i_is_replica>
<i_vstamp>0</i_vstamp>
<r_object_id>5f00000580000103</r_object_id>
</properties>
<audit-data-list>
<audit-data>
    <attr-name>owner_name</attr-name>
    <value-pairs>
        <value-pair>
            <old-value></old-value>
            <new-value>Administrator</new-value>
        </value-pair>
    </value-pairs>
</audit-data>
...
</audit-data-list>
```

```

<links>
  <link rel="self"
    href="http://localhost:8080/dctm-rest/repositories/REPO/audit-trails/
      5f00000580000103"/>
  <link rel="author"
    href="http://localhost:8080/dctm-rest/repositories/REPO/users/
      Administrator"/>
  <link rel="http://identifiers.emc.com/linkrel/audited-object"
    title="document"
    href="http://localhost:8080/dctm-rest/repositories/REPO/documents/
      0900000580000ce9"/>
</links>
</audit-trail>
```



### Example 2-37: JSON

```
{
  "type": "dm_audittrail",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_audittrail",
  "properties": {
    "event_name": "dm_save",
    "event_source": "System Unspecific",
    "r_gen_source": 1,
    "user_name": "Administrator",
    "audited_obj_id": "0900000580000ce9",
    "time_stamp": "2015-03-03T02:34:20.000+00:00",
    "string_1": "Create",
    "id_1": "0000000000000000",
    "id_2": "0000000000000000",
    "id_3": "0000000000000000",
    "id_4": "0000000000000000",
    "id_5": "0000000000000000",
    "chronicle_id": "0900000580000ce9",
    "object_name": "Result.dm_FTQBS_HOURLY",
    "version_label": "1.0",
    "object_type": "dm_document",
    "event_description": "Default Event Set",
    "policy_id": "0000000000000000",
    "workflow_id": "0000000000000000",
    "session_id": "010000058000015b",
    "user_id": "1100000580000102",
    "owner_name": "Administrator",
    "acl_name": "dm_4500000580000101",
    "acl_domain": "Administrator",
    "attribute_list": "acl_name=\"dm_4500000580000101\",
      acl_domain=\"Administrator\",
      world_permit=3,
      group_permit=5,
      group_name=\"docu\",
      owner_permit=7,
      owner_name=\"Administrator\",
      a_storage_type=\"filestore_01\",
      a_full_text=0,
      a_content_type=\"crtext\",
      subject=\"Result of dm_method(dm_FTQBS_HOURLY) with command line:
        -method_verb com.documentum.server.impl.method.fulltext
        .qbs.FTQBSJob -__dm_docbase__ REPO
        -__dm_server_config__ REPO -docbase_name
        R....\",
      title=\"Result of dm_method(dm_FTQBS_HOURLY) with status code
      (0)\",
      object_name=\"Result.dm_FTQBS_HOURLY\",
      attribute_list_id: \"0000000000000000\",
      audit_version": 4,
```



```
        "host_name": "restcs72ga",
        "time_stamp_utc": "2015-03-03T02:34:20.000+00:00",
        "i_audited_obj_class": 0,
        "registry_id": "2600000580000101",
        "i_is_archived": false,
        "audited_obj_vstamp": 0,
        "attribute_list_aspect_id": "0000000000000000",
        "r_object_sequence": 4,
        "i_is_replica": false,
        "i_vstamp": 0,
        "r_object_id": "5f00000580000103"
    },
    "audit-data-list": [
        {
            "attr-name": "owner_name",
            "value-pairs": [
                {
                    "old-value": "",
                    "new-value": "Administrator"
                }
            ]
        },
        {
            "attr-name": "a_content_type",
            "value-pairs": [
                {
                    "old-value": "",
                    "new-value": "crtext"
                }
            ]
        }
        ...
    ],
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails/5f00000580000103"
        },
        {
            "rel": "author",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/users/Administrator"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/audited-object",
            "title": "document",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/documents/0900000580000ce9"
        }
    ]
}
```



#### 2.9.1.4.3 Delete an audit trail

This operation is used to delete an audit trail. Only logged in users with PURGE privileges in their extended user privileges can perform this operation.

##### 2.9.1.4.3.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
DELETE	trace	This parameter determines whether or not this purge operation is traceable.	string	<i>true, false</i>	<i>false</i>

##### 2.9.1.4.3.2 Request media types

N/A

##### 2.9.1.4.3.3 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.9.1.4.3.4 Request body

N/A

##### 2.9.1.4.3.5 Response headers

N/A

##### 2.9.1.4.3.6 Response media types

N/A

##### 2.9.1.4.3.7 Response status

- 204 - Delete successful
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.9.1.4.3.8 Response body

N/A

### 2.9.2 Audit Trails

The Audit Trails resource provides a collection of user operations that occurred within the repository of “[Registered Audit Events](#)” on page 524 and “[Audit Policies](#)” on page 88. This resource allows a logged in user to view their most recent audit trails. However, if a logged in user has `VIEW` privileges in their extended user privileges, then they can view all of the audit trails that are within the repository.

#### 2.9.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

#### 2.9.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Audit Trails
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “[Audit Trail](#)” on page 95 resource
- Entry ID: The Audit Trail resource URI
- Entry Title: The id of the specified Audit Trail
- Entry Updated: The creation date of the specified Audit Trail

### 2.9.2.3 Link relations

Link relation	Description	Resource reference
self	This resource	"Audit Trails" on page 102

### 2.9.2.4 Operations

#### 2.9.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves all of the audit trails in a specified repository.

#### 2.9.2.4.2 Get all audit trails

This operation retrieves all of the audit trails from a specified repository.

##### 2.9.2.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• inline</li> <li>• page</li> <li>• items-per-page</li> <li>• include-total</li> <li>• filter</li> <li>• links</li> <li>• view</li> <li>• sort</li> </ul>	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

##### 2.9.2.4.2.2 Request media types

N/A

#### 2.9.2.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.9.2.4.2.4 Request body

N/A

#### 2.9.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.9.2.4.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.9.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.9.2.4.2.8 Response body

##### ➡ Example 2-38: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/audit-trails</id>
    <title>Audit Trails</title>
    <author>
      <name>OpenText Documentum</name>
    </author>
    <updated>2017-04-11T02:44:37.048+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
      100
    </dm:items-per-page>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/audit-trails"/>
    <link rel="next"
```

```

        href="http://localhost:8080/dctm-rest/repositories/REPO/audit-trails?
            items-per-page=100&page=2" />
    <link rel="first"
        href="http://localhost:8080/dctm-rest/repositories/REPO/audit-trails?
            items-per-page=100&page=1" />
    <entry>
        <id>5f00000580000100</id>
        <title>5f00000580000100</title>
        <summary>Logon Failure</summary>
        <updated>2015-03-03T02:28:26.000+00:00</updated>
        <published>2015-03-03T02:28:26.000+00:00</published>
        <link rel="edit"
            href="http://localhost:8080/dctm-rest/repositories/REPO/audit-trails/
                5f00000580000100" />
        <content type="application/vnd.emc.documentum+xml"
            src="http://localhost:8080/dctm-rest/repositories/REPO/audit-
trails/
                5f00000580000100" />
    </entry>
    ...
</feed>
```



### Example 2-39: JSON

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails",
    "title": "Audit Trails",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2017-04-11T02:43:01.502+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails?
                items-per-page=100&page=2"
        },
        {
            "rel": "first",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails?
                items-per-page=100&page=1"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails/
                5f00000580000100",
            "title": "5f00000580000100",
            "summary": "Logon Failure",
            "updated": "2015-03-03T02:28:26.000+00:00",
            "content": {
                "type": "application/vnd.emc.documentum+xml",
                "src": "http://localhost:8080/dctm-rest/repositories/REPO/audit-
trails/
                    5f00000580000100"
            }
        }
    ]
}
```

```

    "published": "2015-03-03T02:28:26.000+00:00",
    "links": [
      {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/audit-
trails/
          5f00000580000100"
      }
    ],
    "content": {
      "type": "application/vnd.emc.documentum+json",
      "src": "http://localhost:8080/dctm-rest/repositories/REPO/audit-trails/
        5f00000580000100"
    }
  }
  ...
]
}

```

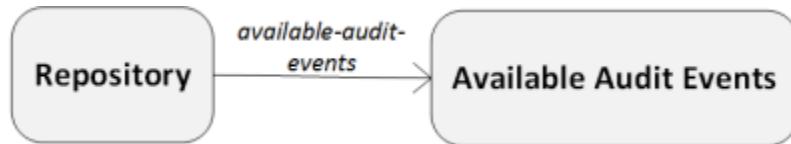


## 2.10 Available Audit Events

The Available Audit Events resource displays all of the audit events that are registered for the specified object, type, or repository. This resource forms part of the audit management resources and closely relates to other resources, such as the Audit Trailresource, Audit Trailsresource, and so on. Logged in users must have CONFIG permissions in their User Extended Privileges to gain access to this resource.

### 2.10.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.10.2 Feed

Is this resource a feed? Yes.

- Feed ID: /repositories/<repositoryName>/available-audit-events
- Feed Title: AvailableAuditEvents
- Feed Updated: Server's current time
- Supports POST?: No

### 2.10.3 Link relations

Link relations	Description	Resource reference
self	This resource	"Available Audit Events" on page 106

### 2.10.4 Operations

#### 2.10.4.1 Supported HTTP methods

Method	Description
GET	GET all available audit events from a specified path.

#### 2.10.4.2 Get all registered audit events

##### 2.10.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameter	Description	Data type	Value range	Default value
GET	links	For more information, see Appendix B, REST Common Definition - URI Request Query Parameters on page 1047.			
	object-id	The ID of the specified object that is requested for its available audit events.	string	Any valid <r_object_id> in Documentum CM Server	null
	type	The name of the type that is requested for its available audit events.	string	Any valid name of a <dm_type> in Documentum CM Server	null



**Note:** When no query parameter, object-ID, or type is specified, a list of the available audit events of the repository are returned.

#### **2.10.4.2.2 Request media types**

N/A

#### **2.10.4.2.3 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.10.4.2.4 Request body**

N/A

#### **2.10.4.2.5 Response headers**

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.10.4.2.6 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### **2.10.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.10.4.2.8 Response body

➡ Example 2-40: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-event-list xmlns="http://identifiers.emc.com/vocab/documentum"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <event-list>
    <event>dm_assume</event>
    <event>dm_authenticate</event>
    <event>dm_connect</event>
    <event>dm_disconnect</event>
    <event>dm_destroy</event>
    <event>dm_getlogin</event>
    <event>dm_logon_failure</event>
    <event>dm_save</event>
    <event>dm_audit</event>
    <event>dm_unaudit</event>
    <event>dm_security_check_failed</event>
  </event-list>
  <links>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/
                available-audit-events/?object-id=1100000580000900" />
    <link rel="http://identifiers.emc.com/linkrel/registered-audit-events"
          href="http://localhost:8080/dctm-rest/repositories/REPO/
                registered-audit-events/?object-id=1100000580000900" />
  </links>
</audit-event-list>
```



➡ Example 2-41: JSON

```
{
  "json-root": "audit-event-list",
  "event-list": [
    "dm_assume",
    "dm_authenticate",
    "dm_connect",
    "dm_disconnect",
    "dm_destroy",
    "dm_getlogin",
    "dm_logon_failure",
    "dm_save",
    "dm_audit",
    "dm_unaudit",
    "dm_security_check_failed"
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                available-audit-events/?object-id=1100000580000900"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/registered-audit-events",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                registered-audit-events/?object-id=1100000580000900"
    }
  ]
}
```



## 2.11 Batches

The **Batches** resource enables you to perform multiple operations using a single HTTP request. For example, you can create 10 cabinets, modify 5 folders, and delete 30 documents using one request. All resources except the **Batches** resource itself and the **Content Media** resource can be embedded in a batch request.



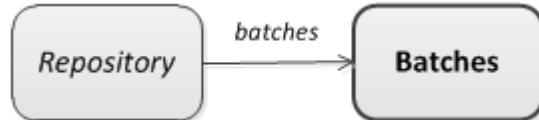
**Note:** The `rest.batch.operations.max.count` property in `rest-api-runtime.properties` determines the maximum number of operations allowed to be embedded in one batch request, which defaults to 1,000. To embed more operations in one batch request, specify a larger value for `rest.batch.operations.max.count`. However, the performance may degrade when the number of operations grows.

In addition to `rest.batch.operations.max.count`, how the system handles batch requests also depends on the following runtime properties in `rest-api-runtime.properties`.

Property	Description	Default
<code>rest.batch.running.tasks.max.count</code>	Specifies the maximum number of batch requests that can be executed simultaneously.  The system rejects new batch requests if the maximum number is reached.	50
<code>rest.batch.core.thread.pool.size</code>	Specifies the size of the thread pool used to execute background tasks.	5
<code>rest.batch.multipart.cache.max.size</code>	Specifies the maximum cache size in megabyte for a multipart batch request.  Setting the value of this property to 0 disables caching for the multipart batch requests.	20

## 2.11.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

## 2.11.2 Feed

No feed support.

## 2.11.3 Link relations

N/A

## 2.11.4 Operations

### 2.11.4.1 Supported HTTP methods

Method	Description
POST	Create and execute a batch request.

### 2.11.4.2 Create and execute a batch request

A POST request to the Batches resource creates a Batch resource, which enables you to perform multiple operations in a single HTTP request. The request contains the details of each embedded operation. Additionally, you can control how the REST server processes the batch request (such as transactional verses, non-transactional verses, sequential verses, and non-sequential verses) by specifying the required properties in the initial request.

#### 2.11.4.2.1 HTTP Request method and query parameters

POST

#### 2.11.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- multipart/related

#### 2.11.4.2.3 Request headers

- Authorization

The Authorization header (user credential token) is applied to all operations within the batch request. Embedded operations in the batch request cannot set their own Authorization headers.

- Accept
- Content-Type
- Accept-Language

If the value is provided in the batch operation, the same value is used in all the requests which are part of the batch operation unless there is no specific value configured for this header at an individual request.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.11.4.2.4 Request body

The request body contains properties of the batch request, which controls how the REST server processes the request along with the properties of each operation that is embedded in the request.

*Batch request properties*

Property	Data type	Description
description	string	Description of the batch request.

Property	Data type	Description
transactional	string	<p>Determines whether to process the batch request as a transaction.</p> <p><i>true</i>: All embedded operations are treated as one atomic unit, meaning that either all operations within a transaction are performed, or none of the operations are performed. A transactional batch request ensures that operations made by one process are not interrupted by another.</p> <p><i>false</i>: Embedded operations are not treated as one atomic unit. The batch operation may continue or terminate, depending on the <code>on-error</code> setting (if <code>sequential</code> is set to <i>true</i>), when one or more operations within the batch fail.</p> <p>Default value: <i>true</i></p>
sequential	string	<p>Determines whether to process the batch request sequentially for non-transactional batch requests.</p> <p><i>true</i>: Operations in a batch are executed in the submission order.</p> <p><i>false</i>: The REST server determines whether to execute the operations in sequence or in parallel.</p> <p>Default value: <i>true</i></p>
on-error	string	<p>Determines whether to continue or terminate a batch operation when one or more of the embedded operations fail.</p> <p>This property works when <code>transactional</code> is set to <i>false</i> and <code>sequential</code> is set to <i>true</i>.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>CONTINUE</b>: The REST server continues processing the batch operation when one or more of the embedded operations fail.</li> <li>• <b>FAIL</b>: The REST server terminates the batch operation when one or more of the embedded operations fail.</li> </ul> <p>Default value: FAIL</p>
return-request	string	<p>Determines whether to return the request for each operation.</p> <p>Default value: <i>false</i></p>

### Operation properties

Each batch request must have one operation at least. The maximum number of operations allowed to be embedded in one request is limited by the `rest.batch.operations.max.count` parameter.

Property	Data type	Description
id *	string	<p>Operation identifier.</p> <p>Each embedded operation must have its unique identifier in one batch request. REST services does not have constraints on how this value is produced. For example, you can use ordering numbers, UUIDs, or even random numbers.</p>
description	string	Description of the operation.
request *	string	<p>HTTP method and target URI of the operation.</p> <p>The URI can be either an absolute URI you get from a certain response or a relative URI, which starts with a slash (/).</p> <p>Example (absolute URI):</p> <pre>&lt;request method="POST" uri="http://localhost:8080/repositories/REPO/folders/0c00208080000107/objects"&gt;</pre> <p>Example (relative URI):</p> <pre>&lt;request method="POST" uri="/repositories/REPO/folders/0c00208080000107/objects"&gt;</pre> <p>The repository in the URI must be the same repository where the Batches resource is located.</p>
Content-Type header	string	<p>MIME type of the request body.</p> <p>Example: &lt;header name="Content-Type" value="application/vnd.emc.documentum+xml"/&gt;</p>
Accept header	string	Acceptable media type for the response. Example: <header name="Accept" value="application/vnd.emc.documentum+xml"/>

Property	Data type	Description
Entity	CDATA/ Escaped String	<p>Request body of the operation.</p> <p>For XML representation, the request body is presented in a CDATA block or escaped.</p> <pre>&lt;entity&gt;   &lt;![CDATA[     &lt;properties&gt;       &lt;object_name&gt;readme&lt;/object_name&gt;       &lt;title&gt;Readme for indexing&lt;/title&gt;     &lt;/properties&gt;   &lt;/document&gt; ]]&gt; &lt;/entity&gt;</pre> <p>For JSON representation, the request body is presented as a string.</p> <pre>{   "entity" : "{\"properties\":{\"object_name\":\"my test object\"}}" }</pre>
Yes *		

#### 2.11.4.2.4.1 Request samples

##### Example 2-42: XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<batch xmlns="http://identifiers.emc.com/vocab/documentum"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <description>batch for indexing</description>
  <transactional>false</transactional>
  <sequential>false</sequential>
  <on-error>CONTINUE</on-error>
  <return-request>true</return-request>
  <operations>
    <operation id="operation_1">
      <description>create a document operation</description>
      <request method="PUT" uri="/objects">
        <header name="Content-Type" value="application/vnd.emc.documentum
+xml"/>
        <header name="Accept" value="application/vnd.emc.documentum+xml"/>
        <entity>
          <![CDATA[
            <document type="dm_document">
              <properties>
                <object_name>readme</object_name>
                <title>Readme for indexing</title>
              </properties>
            </document>
          ]]>
        </entity>
      </request>
    </operation>
    ...
  <operations>
</batch>
```



 **Example 2-43: JSON representation**

```
{
  "transactional" : true,
  "sequential" : false,
  "on-error" : "CONTINUE",
  "return-request" : true,
  "operations" : [ {
    "id" : "id-001",
    "description" : "create a document",
    "request" : {
      "method" : "POST",
      "uri" : "/<base URI>/objects",
      "headers" : [ {
        "name" : "Content-Type",
        "value" : "application/vnd.emc.documentum+json"
      }],
      "entity" : "{\"properties\":{\"object_name\":\"my test object\"}}"
    }
  }]
}
```



#### 2.11.4.2.4.2 Multipart request

Operations embedded in a batch request may have binary contents attached, such as content creation, and content checkin. Foundation REST API leverages the XML-binary Optimized Packaging (XPO) protocol to support multipart operations in batches. As defined in XPO, a multipart batch request must have the `Multipart/Related Content-Type` header.

Typically, a multipart batch request contains several portions as shown in the following pattern:

```
-- boundary
Content-Type: ...
Content-ID: ...
Content-disposition: ...
<batch>
  <operations>
    <operation id="id-1">
      <request>
        ...
        <entity>
          ...
        </entity>
        <attachment>
          <xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/include'
            href='cid:id-1-content' />
        </attachment>
      </request>
    </operation>
    <operation id="id-2">
      <request>
        ...
        <entity>
          ...
        </entity>
      </request>
    </operation>
  </operations>
</batch>
```

```

</entity>
<attachment>
<xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/
include'
    href='cid:id-2-content' />
</attachment>
</request>
</operation>
...
</operations>
</batch>
-- boundary
Content-Type: ...
Content-ID: id-1-content
Content-disposition: ...
Attachment 1
-- boundary
Content-Type: ...
Content-ID: id-2-content
Content-disposition: ...
Attachment 2
-- boundary
...
-- boundary --

```

The `batch` element is identical to its counterpart in standard batch requests with the only addition of an extra `attachment` element in the `request` block of each embedded operation. The `attachment` element contains a content ID (`cid`) specifying the binary content to be attached which is defined in the subsequent portions of the batch request. The sequence of the portions holding binary contents must be identical to the sequence they are referenced in the `batch` element.

#### *Multipart/Related content-type header*

The Multipart/Related Content-Type header contains the following parameters:

Parameters	Descriptions	Required
boundary	Specifies the boundary used to separate each portion	Yes
type	Specifies the media type of the batch request. Valid values are: <ul style="list-style-type: none"> <li>• application/xop+xml (for XML)</li> <li>• application/jop+json (for JSON)</li> </ul>	Yes
start	Specified the portion to be processed first by Content ID. If not specified, the first portion is processed first. In a multipart batch request, the start portion must be a <code>batch</code> block.	No
start-info	The media type of the start portion. Valid values are: <ul style="list-style-type: none"> <li>• application/vnd.emc.documentum+xml (for XML)</li> <li>• application/vnd.emc.documentum+json (for JSON)</li> </ul>	Yes

#### *Portion header set*

A portion header set contains the following headers:

Headers	Descriptions	Required
Content-Type	<p>For the start portion batch, this parameter specifies the media type of the batch request together with the media type of the start portion in the <code>type</code> property, for example:</p> <pre>Content-Type: application/xop+xml; type="application/vnd.emc.documentum+xml"</pre> <p>For portions holding binary contents, this parameter specifies the media type, for example:</p> <pre>Content-Type: text/plain</pre>	Yes
Content-ID	Specifies the Content ID of the portion.	Yes
Content-disposition	<p>This parameter must be set with the following pattern:</p> <pre>Content-disposition: form-data; name=&lt;Content ID&gt;</pre> <p>Example: Content-disposition: form-data; name=id-01-content</p>	Yes

#### *Headers in operation blocks versus headers in binary content blocks*

Headers in an operation block describe the entity block next to it. Typically, the header set includes a Content-Type header indicating the media type of the content in the entity block and an Accept header indicating the expected media type of the response. A header set in a binary content block also contains a Content-Type header. This Content Type header indicates media type of the binary content.

The following excerpt illustrates a multipart request that contains both an entity block and binary content:

```
...
<operation id="id-100">
    <request method="POST"
        uri="/repositories/REPO/folders/0c00208080000107/objects">
        <header name="Content-Type" value="application/vnd.emc.documentum+xml"/>
    >
    <header name="Accept" value="application/vnd.emc.documentum+xml"/>
    <entity>
        <object><properties><object_name>my-test-doc</object_name>
            </properties></object>
        </entity>
        <attachment><xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/include'
            href='cid:id-100-content' /></attachment>
    </request>
</operation>
...
--frontier
Content-Type: text/plain
Content-ID: id-100-content
Content-disposition: form-data; name=id-100-content
```

This is the content of id-100

In a scenario where a multipart request does not include an entity block (for example, you upload binary content without updating any resource property), headers in operation blocks are used to describe the binary content. In the binary content block, all headers except Content-ID and Content-disposition are ignored. As a result, you must specify the media type of the binary content in the Content-Type header of the operation block.

In the following example, the media type of this individual request is `text/plain` and the REST server returns the response in `application/vnd.emc.documentum+xml` as specified in the operation block.

```
<operation id="id-100">
    <request method="POST"
        uri="/repositories/REPO/folders/0c00208080000107/objects">
        <header name="Content-Type" value="text/plain"/>
        <header name="Accept" value="application/vnd.emc.documentum+xml"/>
        <attachment><xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/include'
            href='cid:id-100-content' /></attachment>
    </request>
</operation>

--frontier
Content-Type: text/html
Accept: text/application/vnd.emc.documentum+json
Content-ID: id-100-content
Content-disposition: form-data; name=id-100-content

This is the content of id-100
```

In a Multipart/Related batch request, you are also allowed to embed requests that do not contain attachments. In this scenario, the headers in the operation block describe the entity block next to them and no binary content block pertains to the Request.

```
<operation id="id-100">
    <request method="POST"
        uri="/repositories/REPO/folders/0c00208080000107/objects">
        <header name="Content-Type" value="application/vnd.emc.documentum+xml"/>
        <header name="Accept" value="application/vnd.emc.documentum+xml"/>
        <entity>
            <object><properties><object_name>my-test-doc</object_name>
                </properties></object>
        </entity>
    </request>
</operation>
```

## Multipart request samples

### Example 2-44: Request in application/xop+xml

```
Request header:
{Multipart/Related;boundary=frontier;
type="application/xop+xml";
start="batch";
```

```
start-info="application/vnd.emc.documentum+xml"}  
  
Payload:  
--frontier  
Content-Type: application/xop+xml; type="application/vnd.emc.documentum+xml"  
Content-ID: batch  
Content-disposition: form-data; name=batch  
  
<?xml version="1.0" encoding="UTF-8"?>  
<batch>  
    <transactional>true</transactional>  
    <sequential>true</sequential>  
    <operations>  
        <operation id="id-100">  
            <request method="POST"  
uri="/repositories/REPO/folders/0c00208080000107/objects">  
                <header name="Content-Type" value="application/vnd.emc.documentum+xml"/>  
>  
                <header name="Accept" value="application/vnd.emc.documentum+xml"/>  
                <entity>  
                    <object><properties><object_name>my-doc1</object_name>  
                        </properties></object>  
                    </entity>  
                    <attachment><xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/include'  
href='cid:id-100-content' /></attachment>  
                </request>  
            </operation>  
            <operation id="id-101">  
                <request method="POST" uri="/repositories/REPO/folders/0c00208080000107/objects">  
                    <header name="Content-Type" value="application/vnd.emc.documentum+xml"/>  
>  
                    <header name="Accept" value="application/vnd.emc.documentum+xml"/>  
                    <entity>  
                        <object><properties><object_name>my-doc2</object_name>  
                            </properties></object>  
                        </entity>  
                        <attachment><xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/include'  
href='cid:id-101-content' /></attachment>  
                    </request>  
                </operation>  
            </operations>  
        </batch>  
  
--frontier  
Content-Type: text/plain  
Content-ID: id-100-content  
Content-disposition: form-data; name=id-100-content  
This is the content of id-100  
  
--frontier  
Content-Type: text/plain  
Content-ID: id-101-content  
Content-disposition: form-data; name=id-101-content  
This is the content of id-101  
--frontier--
```



▶ **Example 2-45: Request in application/jop+json**

```

Request header:
{Multipart/Related;boundary=frontier;
type="application/jop+json";
start="batch";
start-info="application/vnd.emc.documentum+json"}

Payload:
--frontier
Content-Type: application/jop+json; type="application/vnd.emc.documentum
+json"
Content-ID: batch
Content-disposition: form-data; name=batch
{
  "return-request" : true,
  "operations" :
  [
    {
      "id" : "id-100",
      "description" : "create object with content",
      "request" :
      {
        "method" : "POST",
        "uri" : "/repositories/REPO/folders/0c00208080000107/objects",
        "headers" :
        [
          {
            "name" : "Content-Type",
            "value" : "application/vnd.emc.documentum+json"
          },
          {
            "name" : "Accept",
            "value" : "application/vnd.emc.documentum+json"
          }
        ],
        "entity" : "{\"properties\":{\"object_name\":\"my test object\"}}",
        "attachment" : {
          "Include" : {
            "href" : "cid:id-100-content"
          }
        }
      },
      {
        "id" : "id-101",
        "description" : "create object with content",
        "request" :
        {
          "method" : "POST",
          "uri" : "/repositories/REPO/folders/0c00208080000107/objects",
          "headers" :
          [
            {
              "name" : "Content-Type",
              "value" : "application/vnd.emc.documentum+json"
            },
            {
              "name" : "Accept",
              "value" : "application/vnd.emc.documentum+json"
            }
          ]
        }
      }
    }
  ]
}

```

```
        ],
        "entity" : "{\"properties\":{\"object_name\":\"my test object\"}}",
        "attachment" : {
            "Include" : {
                "href" : "cid:id-101-content"
            }
        }
    }
]

--frontier
Content-Type: text/plain
Content-ID: id-100-content
Content-disposition: form-data; name=id-100-content
i'm the content of id-100

--frontier
Content-Type: text/plain
Content-ID: id-101-content
Content-disposition: form-data; name=id-101-content
i'm the content of id-101
--frontier--
```



#### 2.11.4.2.5 Response headers

- Content-Type

#### 2.11.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.11.4.2.7 Response status

- 201 - Batch executed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.11.4.2.8 Exceptions

Error Code	Description
E_ILLEGAL_TASK_OPERATIONS	The number of operations embedded in the batch request exceeds the threshold specified in <code>rest.batch.operations.max.count</code> .
E_BATCH_NO_OPERATION	No operation in the request.
E_BATCH_NO_OPERATION_ID	One or more operations in the batch request do not contain an identifier.
E_BATCH_OPERATION_ID_DUPLICATED	The batch request contains duplicated operation identifiers.
E_BATCH_NO_OPERATION_REQUEST	One or more operations in the batch request do not have a request defined.
E_BATCH_NO_REQUEST_URI	One or more operations in the batch request do not have a URI defined.
E_BATCH_NO_REQUEST_METHOD	One or more operations in the batch request do not have a method defined.
E_METHOD_NOT_ALLOWED	One or more operations in the batch request do not have an invalid method defined.
E_BATCH_REQUEST_INVALID_URI	The request URI is invalid.
E_BATCH_REQUEST_URI_ROOT	The relative URI does not start with "/".
E_BATCH_PROHIBITION	There are operations on resources with the <code>BatchProhibition</code> annotation defined.
E_BATCH_TRANSACTION_PROHIBITION	There are resources with the <code>TransactionProhibition</code> annotation defined in a transactional batch request.

#### 2.11.4.2.9 Response body

In this release, batch requests are executed in synchronous mode. The response is not returned until the REST server completes the request. The response of a batch request contains the following:

- Batch properties. In addition to those described in [Batch Request Properties on page 112](#), the response also includes:
  - state: Indicates the state of the batch operation. In synchronous mode, the value of this property is always `Finished`.
  - sub-state: Indicates the sub-state of a finished batch operation. Valid values are:
    - failed: The REST server completes the batch operation. However, all the operations fail. Generally, this value is returned for transactional batch requests.

- `finished_with_error`: The REST server completes the batch operation, while the operation is partially failed. Generally, this value is returned for non-transactional batch requests.
- `owner`: Indicates the user submitting the batch request.
- `submitted`: Indicates the time point when the batch request is submitted.
- `started`: Indicates the time point when the server starts processing the batch request.
- `finished`: Indicates the time point when the batch operation is completed.
- Each embedded request if `return-request` is *true*.
- The response of each embedded operation.

#### ➤ Example 2-46: XML representation

```

<?xml version="1.0" encoding="UTF-8"?>
<batch xmlns="http://identifiers.emc.com/vocab/documentum"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <transactional>true</transactional>
    <sequential>false</sequential>
    <on-error>CONTINUE</on-error>
    <return-request>true</return-request>
    <state>FINISHED</state>
    <submitted>2014-07-03T16:37:32.147+08:00</submitted>
    <started>2014-07-03T16:37:32.147+08:00</started>
    <finished>2014-07-03T16:37:32.246+08:00</finished>
    <owner>dmadmin</owner>
    <operations>
        <operation id="id-100">
            <description>get users</description>
            <state>FINISHED</state>
            <started>2014-07-03T16:37:32.147+08:00</started>
            <finished>2014-07-03T16:37:32.158+08:00</finished>
            <request method="GET" uri="/repositories/REPO/users?items-per-
page=1">
                <header name="Content-Type" value="application/
vnd.emc.documentum+xml"/>
                    <header name="Accept" value="application/atom+xml"/>
                </request>
                <response status="200">
                    <header name="Content-Type" value="application/atom
+xml; charset=UTF-8"/>
                    <entity><?xml version="1.0" encoding="UTF-8"?><feed
xmlns="http:
//www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-inst
ance" xmlns:dm="http://identifiers.emc.com/vocab/
documentum"><id>
http://localhost:8080/dctm-rest/repositories/REPO/users</
id><titl
e>Users</title><author><name>OpenText Documentum</name&
gt;</author><dm:page>1</dm:page><dm:items-per-page>
1</dm:items-per-page><link rel="self" href="http://localhost:
8080/
dctm-rest/repositories/REPO/users?items-per-page=1"/><link
rel="next"
href="http://localhost:8080/dctm-rest/repositories/REPO/users?
items-per-page

```

```

dctm-rest          =1&page=2"/><link rel="first" href="http://localhost:8080/
                  /repositories/REPO/users?items-per-page=1&page=1" /
><entry><id      >http://localhost:8080/dctm-rest/repositories/REPO/users/
Administrator<   /id><title>Administrator</title><updated>2014-01-21T12:24
                  :16.000+08:00</updated><summary/><content type="application/vnd
                  .emc.documentum+xml" src="http://localhost:8080/dctm-rest/
repositories/REPO/us   ers/Administrator"/><link rel="edit" href="http://localhost:
8080/dctm-res     t/repositories/REPO/users/Administrator"/></entry></feed>
                  </entity>
                </response>
              </operation>
            <operation id="id-1001">
              <description>get cabinets</description>
              <state>FINISHED</state>
              <started>2014-07-03T16:37:32.158+08:00</started>
              <finished>2014-07-03T16:37:32.244+08:00</finished>
              <request method="GET" uri="/repositories/REPO/cabinets?items-per-
page=1">
                <header name="Content-Type" value="application/
vnd.emc.documentum+xml"/>
                <header name="Accept" value="application/atom+xml"/>
              </request>
              <response status="200">
                <header name="Content-Type" value="application/atom
+xml; charset=UTF-8"/>
                <entity><?xml version="1.0" encoding="UTF-8"?><feed
xmlns="http://
www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
                  xmlns:dm="http://identifiers.emc.com/vocab/
documentum"><id>http://
localhost:8080/dctm-rest/repositories/REPO/cabinets</id><title>
Cabinets</title><updated>2014-07-03T16:37:32.238+08:00</updat
ed><author><name>OpenText Documentum</name></author>&lt
;dm:page>1</dm:page><dm:items-per-page>1</dm:items-per-page
><link rel="self" href="http://localhost:8080/dctm-rest/
repositories/RE
PO/cabinets?items-per-page=1"/><link rel="next" href="http://
localhost:8080/dctm-rest/repositories/REPO/cabinets?items-per-
page=1&page=2"/><
link rel="first" href="http://localhost:8080/dctm-rest/
repositories/REPO/cabine
ts?items-per-page=1&page=1"/><entry><id>http://localhost:
808
0/dctm-rest/repositories/REPO/objects/0c0020808000dfc6</
id><title>" + cabinet + "</title><updated>2014-06-30T16:13:52.000+08:00</
upd
ated><summary>dm_cabinet 0c0020808000dfc6</summary><author>
<name>dmadmin</name><uri>http://localhost:8080/dctm-rest/repos
itories/REPO/users/dmadmin</uri></author><content
type="application/vnd.emc.documentum+xml" src="http://localhost:8080/dctm-
rest/repositories/REP
O/cabinets/0c0020808000dfc6"/><link rel="edit" href="http://
localhost:8080/

```

```

entry></fee
      dctm-rest/repositories/REPO/cabinets/0c0020808000dfc6"/><
entry></fee
      d></entity>
</response>
</operation>
</operations>
</batch>
```



#### Example 2-47: JSON representation

```
{
  "transactional" : true,
  "sequential" : false,
  "on-error" : "CONTINUE",
  "return-request" : true,
  "state" : "FINISHED",
  "submitted" : "2014-07-03T16:34:45.989+08:00",
  "started" : "2014-07-03T16:34:45.989+08:00",
  "finished" : "2014-07-03T16:34:46.107+08:00",
  "owner" : "dmadmin",
  "operations" : [ {
    "id" : "id-100",
    "description" : "get users",
    "state" : "FINISHED",
    "started" : "2014-07-03T16:34:45.989+08:00",
    "finished" : "2014-07-03T16:34:46.020+08:00",
    "request" : {
      "name" : "request",
      "method" : "GET",
      "uri" : "/repositories/REPO/users?items-per-page=1",
      "headers" : [ {
        "name" : "Content-Type",
        "value" : "application/vnd.emc.documentum+json"
      }, {
        "name" : "Accept",
        "value" : "application/vnd.emc.documentum+json"
      } ]
    },
    "response" : {
      "name" : "response",
      "status" : 200,
      "headers" : [ {
        "name" : "Content-Type",
        "value" : "application/vnd.emc.documentum+json; charset=UTF-8"
      }],
      "entity" : "{\\"id\\":\\"http://localhost:8080/dctm-rest/repositories/REPO/
users
      \\", \"title\": \"Users\", \"author\": [{\"name\": \"OpenText Documentum\"}],
      \\\"page\\\":1,
      \"items-per-page\\\":1, \"links\\\": [{\"rel\": \"self\", \"href\": \"http://
localhost:8
      080/dctm-rest/repositories/REPO/users?items-per-page=1\"}, {\"rel\": \"next
      \", \"h
      ref\": \"http://localhost:8080/dctm-rest/repositories/REPO/users?items-
      per-page=
      1&page=2\"}, {\"rel\": \"first\", \"href\": \"http://localhost:8080/dctm-
      rest/reposi
      tories/REPO/users?items-per-page=1&page=1\"}], \"entries\\\": [{\"id\": :
```

```

\"http://localhost:8080/dctm-rest/repositories/REPO/users/Administrator\",
  \"title\": \"Administrator\",
  \"updated\": \"2014-01-21T12:24:16.000+08:00\",
  \"summary\": \"\",
  \"content-type\": \"application/vnd.emc.documentum+json\",
  \"src\": \"http://localhost:8080/dctm-rest/repositories/REPO/users/Administrator\",
  \"links\": [
    {
      \"rel\": \"edit\",
      \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/users/Administrator\"
    }
  ],
  {
    \"id\": \"id-101\",
    \"description\": \"get cabinets\",
    \"state\": \"FINISHED\",
    \"started\": \"2014-07-03T16:34:46.020+08:00\",
    \"finished\": \"2014-07-03T16:34:46.105+08:00\",
    \"request\": {
      \"name\": \"request\",
      \"method\": \"GET\",
      \"uri\": \"/repositories/REPO/cabinets?items-per-page=1\",
      \"headers\": [
        {
          \"name\": \"Content-Type\",
          \"value\": \"application/vnd.emc.documentum+json\"
        },
        {
          \"name\": \"Accept\",
          \"value\": \"application/vnd.emc.documentum+json\"
        }
      ],
      \"response\": {
        \"name\": \"response\",
        \"status\": 200,
        \"headers\": [
          {
            \"name\": \"Content-Type\",
            \"value\": \"application/vnd.emc.documentum+json; charset=UTF-8\"
          }
        ],
        \"entity\": {
          \"id\": \"http://localhost:8080/dctm-rest/repositories/REPO/cabinets\",
          \"title\": \"Cabinets\",
          \"updated\": \"2014-07-03T16:34:46.103+08:00\",
          \"author\": [
            {
              \"name\": \"OpenText Documentum\"
            }
          ],
          \"links\": [
            {
              \"rel\": \"self\",
              \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/cabinets?items-per-page=1\"
            },
            {
              \"rel\": \"next\",
              \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/cabinets?items-per-page=1&page=2\"
            },
            {
              \"rel\": \"first\",
              \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/cabinets?items-per-page=1&page=1\"
            }
          ],
          \"entries\": [
            {
              \"id\": \"http://localhost:8080/dctm-rest/repositories/REPO/objects/0c0020808000dfc6\",
              \"title\": \"\",
              \"updated\": \"2014-06-30T16:13:52.000+08:00\",
              \"summary\": \"dm_cabinet 0c0020808000dfc6\",
              \"author\": [
                {
                  \"name\": \"dadmin\",
                  \"uri\": \"http://localhost:8080/dctm-rest/repositories/REPO/users/dadmin\"
                }
              ],
              \"content-type\": \"application/vnd.emc.\"
            }
          ]
        }
      }
    }
  ]
}

```

```
documentum+json\", \"sr
    c\":\"http://localhost:8080/dctm-rest/repositories/REPO/cabinets/
0c0020808000dfc6\"
        ,\"links\":[{\\"rel\":\"edit\", \\"href\":\"http://localhost:8080/dctm-rest/
repository
            es/REPO/cabinets/0c0020808000dfc6\"}]}]}}
}
}
}
```

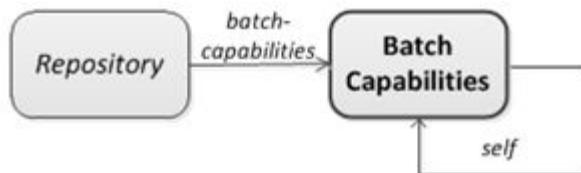


## 2.12 Batch Capabilities

The Batch Capabilities resource indicates what batch features are implemented in Foundation REST API. This includes the supported batch properties, such as transactional/non-transactional, the list of batchable resources, and the list of non-batchable resources.

### 2.12.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

## 2.12.2 Feed

No feed support.

### 2.12.3 Link relations

Link relation	Description	Resource reference
self	The Batch Capabilities resource	<a href="#">"Batch Capabilities" on page 128</a>

## 2.12.4 Operations

### 2.12.4.1 Supported HTTP methods

Method	Description
GET	Retrieve supported batch properties, the list of batchable resources, and the list of non-batchable resources.

### 2.12.4.2 Get Batch Capabilities

Retrieves supported batch properties, the list of batchable resources, and the list of non-batchable resources.

#### 2.12.4.2.1 HTTP Request method and query parameters

GET

#### 2.12.4.2.2 Request media types

N/A

#### 2.12.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.12.4.2.4 Request body

N/A

#### 2.12.4.2.5 Response headers

- Content-Type

#### 2.12.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.12.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.12.4.2.8 Response body

XML or JSON representation that describes the supported batch features and provides the lists of batchable and non-batchable resources.

##### Example 2-48: XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<batch-capabilities xmlns="http://identifiers.emc.com/vocab/documentum"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <transactions>both</transactions>
    <sequence>both</sequence>
    <on-error>both</on-error>
    <batchable-resources>
        <batchable-resource>batch-capabilities</batchable-resource>
        <batchable-resource>cabinet</batchable-resource>
        <batchable-resource>cabinets</batchable-resource>
        <batchable-resource>checked-out-objects</batchable-resource>
        <batchable-resource>content</batchable-resource>
        <batchable-resource>contents</batchable-resource>
        <batchable-resource>current-user</batchable-resource>
        <batchable-resource>current-version</batchable-resource>
        <batchable-resource>default-folder</batchable-resource>
        <batchable-resource>document</batchable-resource>
        <batchable-resource>dql-query</batchable-resource>
        <batchable-resource>folder</batchable-resource>
        <batchable-resource>folder-child-links</batchable-resource>
        <batchable-resource>folder-child-objects</batchable-resource>
        <batchable-resource>folder-link</batchable-resource>
        <batchable-resource>folder-parent-links</batchable-resource>
        <batchable-resource>format</batchable-resource>
        <batchable-resource>formats</batchable-resource>
        <batchable-resource>group</batchable-resource>
        <batchable-resource>group-members</batchable-resource>
        <batchable-resource>groups</batchable-resource>
        <batchable-resource>home-document</batchable-resource>
        <batchable-resource>lock</batchable-resource>
        <batchable-resource>network-location</batchable-resource>
        <batchable-resource>network-locations</batchable-resource>
        <batchable-resource>object</batchable-resource>
        <batchable-resource>product-info</batchable-resource>
        <batchable-resource>relation</batchable-resource>
        <batchable-resource>relation-type</batchable-resource>
        <batchable-resource>relation-types</batchable-resource>
        <batchable-resource>relations</batchable-resource>
        <batchable-resource>repositories</batchable-resource>
        <batchable-resource>repository</batchable-resource>
        <batchable-resource>search</batchable-resource>
        <batchable-resource>type</batchable-resource>
        <batchable-resource>types</batchable-resource>
        <batchable-resource>user</batchable-resource>
        <batchable-resource>users</batchable-resource>
```

```

        <batchable-resource>versions</batchable-resource>
    </batchable-resources>
    <non-batchable-resources>
        <non-batchable-resource>batches</non-batchable-resource>
        <non-batchable-resource>content-media</non-batchable-resource>
    </non-batchable-resources>
    <links>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/
            batch-capabilities.xml" rel="self"/>
    </links>
</batch-capabilities>
```



#### ➤ Example 2-49: JSON representation

```
{
    "transactions" : "both",
    "sequence" : "both",
    "on-error" : "both",
    "batchable-resources" : [
        "batch-capabilities",
        "cabinet", "cabinets",
        "checked-out-objects",
        "content",
        "contents",
        "current-user",
        "current-version",
        "default-folder",
        "document",
        "dql-query",
        "folder",
        "folder-child-links",
        "folder-child-objects",
        "folder-link",
        "folder-parent-links",
        "format",
        "formats",
        "group",
        "group-members",
        "groups",
        "home-document",
        "lock",
        "network-location",
        "network-locations",
        "object",
        "product-info",
        "relation",
        "relation-type",
        "relation-types",
        "relations",
        "repositories",
        "repository",
        "search",
        "type",
        "types",
        "user",
        "users",
        "versions" ],
    "non-batchable-resources" : [
        "batches",
```

```

    "content-media" ],
"links" : [ {"rel" : "self",
            "href" : "http://localhost:8080/dctm-rest/repositories/
REPO/batch-capabilities" } ]}
```



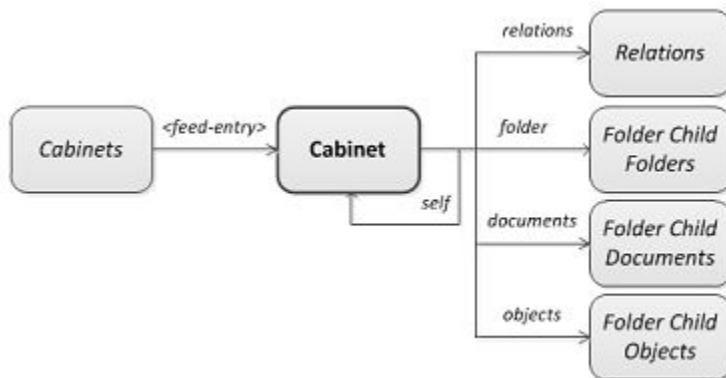
## 2.13 Cabinet (s)

### 2.13.1 Cabinet

The Cabinet resource represents a cabinet in the repository.

#### 2.13.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Cabinets” on page 138 and About the Diagram on page 13.

#### 2.13.1.2 Feed

No feed support.

#### 2.13.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this cabinet.	“Cabinet” on page 132
folders [1]	Collection of folders linked to this cabinet.	“Folder Child Folders” on page 268
relations [1]	Collection of relations related to this cabinet.	“Cabinets” on page 138
documents [1]	Collection of documents associated to this cabinet.	“Folder Child Documents” on page 240

Link relation	Description	Resource reference
objects [1]	Collection of objects of this cabinet.	" <a href="#">Folder Child Objects</a> " on page 282
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

## 2.13.1.4 Operations

### 2.13.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves properties, and other information of the Cabinet resource.
POST	Updates the properties for the Cabinet resource.
DELETE	Deletes the Cabinet resource from a repository.

### 2.13.1.4.2 Get a Cabinet

GET properties, and other information of the Cabinet resource. Properties are returned as embedded elements in the response message body. Other information, such as relationships, is referenced from the link relations of the response message body.

#### 2.13.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			
	links				
	check-subscription	Determines whether or not to check if the current user is subscribed to the requested instance.	string	true, false	false

#### **2.13.1.4.2.2 Request media types**

N/A

#### **2.13.1.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.13.1.4.2.4 Request body**

N/A

#### **2.13.1.4.2.5 Response headers**

- Content-Type

#### **2.13.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.13.1.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.13.1.4.2.8 Response body**

XML or JSON message that represents the cabinet.

#### 2.13.1.4.3 Update a Cabinet

Update the cabinet with given properties. If the client provides undefined properties in the request body, the server throws an exception.

##### 2.13.1.4.3.1 HTTP Request method and query parameters

POST

##### 2.13.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.13.1.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.13.1.4.3.4 Request body

XML or JSON representation of the Cabinet resource. Only properties that can be updated should be put in the message body.

##### 2.13.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.13.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.13.1.4.3.7 Response status

- 200 - Cabinet updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.13.1.4.3.8 Response body

XML or JSON representation of the Cabinet resource. The successful response contains all properties of the updated cabinet.

#### 2.13.1.4.4 Delete a cabinet

Deletes the Cabinet resource from the repository.

If the deletion fails, an exception is thrown and the Cabinet resource (including the cabinet tree and version history) is reverted to the original state.

#### 2.13.1.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	del-non-empty	<p>Specifies whether this operation deletes a non-empty cabinet or not</p> <ul style="list-style-type: none"><li>• <i>true</i> - The cabinet and all its descendants are deleted.</li><li>• <i>false</i> - Only this cabinet is deleted. Deleting a non-empty cabinet results in an error.</li></ul> <p>(boolean)</p>	string	<i>true</i> , <i>false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	del-all-links	<p>Specifies whether a multi-linked descendant is deleted or unlinked from this specified cabinet.</p> <ul style="list-style-type: none"> <li><i>true</i> - This operation deletes a multi-linked descendant along with all its folder links.</li> <li><i>false</i> - This operation only deletes a descendant's link to this cabinet. The descendant is still linked to other folders.</li> </ul> <p> <b>Note:</b> Virtual documents are not supported.</p>	string	<i>true, false</i>	<i>false</i>

#### 2.13.1.4.4.2 Request media types

N/A

#### 2.13.1.4.4.3 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.13.1.4.4.4 Request body

N/A

#### 2.13.1.4.4.5 Response headers

N/A

#### 2.13.1.4.4.6 Response media types

N/A

#### 2.13.1.4.4.7 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.13.1.4.4.8 Response body

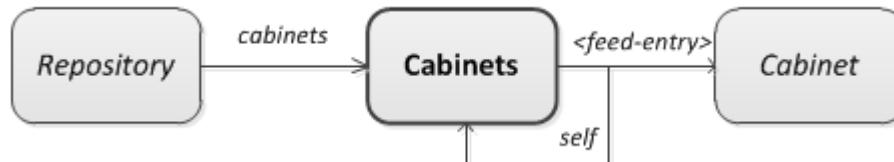
HTTP 204 No Content status upon a successful delete operation. The response body contains no content.

## 2.13.2 Cabinets

The Cabinets resource represents a collection of cabinets in a repository.

### 2.13.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

### 2.13.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the Cabinets resource without the file extension
- Feed Title: List of cabinets
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: “[Cabinet](#)” on page 132
- Entry ID: URI of the cabinet
- Entry Title: Cabinet name
- Entry Summary: Cabinet description
- Entry Updated: r\_modify\_date of the cabinet

### 2.13.2.3 Link relations

Link relation	Description	Resource reference
self	URI for cabinets collection feed	<a href="#">“Cabinets” on page 138</a>
first, last, next, previous	Pagination links	<a href="#">“Cabinets” on page 138</a>

### 2.13.2.4 Operations

#### 2.13.2.4.1 Supported HTTP methods

Method	Description
GET	Lists all available cabinets in the repository for the current user
POST	Create a new cabinet in the repository.

#### 2.13.2.4.2 Get Cabinets

List all available cabinets in the repository for the current user.

##### 2.13.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	object-type	Specifies the exact sub type name to get from the collection of this type.	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	thumbnail	<p>Specifies whether this operation deletes a non-empty folder or not.</p> <ul style="list-style-type: none"><li>• <i>true</i> - Return the thumbnail link for each entry in the collection resource.</li><li>• <i>false</i> - Do not return the thumbnail link for each entry in the collection resource.</li></ul>	string		<i>false</i>

This method supports the following common query parameters:

- inline
- page
- items-per-page
- include-total
- view
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.13.2.4.2.2 Request media types

N/A

#### 2.13.2.4.2.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.13.2.4.2.4 Request body

N/A

#### 2.13.2.4.2.5 Response headers

- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.13.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing)

#### 2.13.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.13.2.4.2.8 Response body

XML or JSON representation of the cabinets collection.

- The body contains a list of cabinets (or subtypes of `dm_cabinet`).
- Each object may contain all properties of the cabinet, depending on the setting of the query parameter `view`.
- The returned child objects collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order. The returned cabinets can be ordered by any non-repeating property.
- The total count is returned only when you explicitly set the `include-total` parameter to `true` in the Request. In this case, the link for last page is returned.
- Each cabinet object must contain links specific to `dm_cabinet` type. See “[Link relations](#)” on page 693 in `SysObject` resource to view the links that are available to the `dm_cabinet` object type.

#### 2.13.2.4.3 Create a cabinet

Create a new cabinet or a subtype of cabinet in the repository.

##### 2.13.2.4.3.1 HTTP Request method and query parameters

POST

##### 2.13.2.4.3.2 Request media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

##### 2.13.2.4.3.3 Request headers

- `Accept`
- `Authorization`
- `Content-Type`

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.13.2.4.3.4 Request body

XML or JSON representation of the cabinet to create.

The `r_object_type` property for the cabinets must be `dm_cabinet` or its subtype, otherwise, the Request is considered as a bad request and rejected. If this property is not specified in a client request, it is set to `dm_cabinet` by default.

#### 2.13.2.4.3.5 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.13.2.4.3.6 Response media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`
- `application/xml`
- `application/json`

#### 2.13.2.4.3.7 Response status

- 201 - Cabinet created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.13.2.4.3.8 Response body

XML or JSON representation of the cabinet created.

## 2.14 Checked Out Objects

The Checked Out Objects resource represents a collection of checked out objects in a repository.

### 2.14.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

### 2.14.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the checked out objects
- Feed Title: Checked out objects
- Feed Updated: Server's current time
- Supports POST: No
- Entry: Each entry represents a checked out object.
- Entry ID: URI of the version
- Entry Title: `object_name` of the object
- Entry Updated: `r_modify_date` for the object

### 2.14.3 Link relations

Link relation	Description	Resource reference
self	This All Versions resource	<a href="#">“Checked Out Objects” on page 143</a>
first, last, next, previous	Pagination links	<a href="#">“Checked Out Objects” on page 143</a>

## 2.14.4 Operations

### 2.14.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of checked out objects.

### 2.14.4.2 Get checked out objects

Retrieve a collection of checked out objects according to the query parameters specified in the request.

#### 2.14.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	checked-out-by	<p>Specifies the lock owner of the checked out objects with the following valid values:</p> <ul style="list-style-type: none"> <li>• current-user</li> <li>• specific-user</li> <li>• any</li> </ul> <p>Only checked out objects with the specified lock owner are returned in the result.</p>	enum	N/A	current-user
	user-name	<p>Specifies the user name of the lock owner.</p> <p>Only works when the checked-out-by parameter is set to specific-user</p>	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	thumbnail	<p>Specifies whether this operation deletes a non-empty folder or not.</p> <ul style="list-style-type: none"> <li>• <i>true</i> - Return the thumbnail link for each entry in the collection resource.</li> <li>• <i>false</i> - Do not return the thumbnail link for each entry in the collection resource.</li> </ul>	string	<i>true, false</i>	<i>false</i>
	include-all-versions	<p>Specifies whether or not to list all versions of the child object's link.</p> <ul style="list-style-type: none"> <li>• <i>true</i> - Return all versions.</li> <li>• <i>false</i> - Only return the current version.</li> </ul>	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	q	<p>Specifies the search criterion with a full-text expression in Simple Search Language.</p> <p>Parameter q must be encoded because it may contain non-English locale characters.</p> <p> <b>Note:</b> International characters that are used in this query parameter must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.</p>	string	N/A	No search criteria is used for the search.

This method supports the following common query parameters:

- inline
- page
- items-per-page

- include-total
- links
- filter
- view

The q parameter is supported for full text searching with a subset of the Simple Search Language, however the parenthesis is not supported.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### **2.14.4.2.2 Request media types**

N/A

#### **2.14.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.14.4.2.4 Request body**

N/A

#### **2.14.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.14.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.14.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.14.4.2.8 Response body

XML or JSON representation of the checked out objects.

- Each object may contain all or a set of properties of the objects, depending on the setting of the query parameter `view`.
- The returned child objects collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order by object name.

## 2.15 Checksum

Clients can retrieve the checksum using the checksum endpoint for an object ID.

### 2.15.1 Feed

No feed support.

### 2.15.2 Link relations

No link relations.

### 2.15.3 Operations

#### 2.15.3.1 Supported HTTP methods

Only GET method is supported to fetch the checksum of an object.

#### 2.15.3.2 Get the checksum

GET the checksum of an object using the `repositories/{{repository-name}}/objects/{{object-id}}/checksum` endpoint.

**2.15.3.2.1 Request method and query parameters**

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	format	Specifies the format of contents of an object to which checksum is required.  For example: format=<cr text> or <pdf>	string	string	null
	page	Specifies the content page number with any non-negative number adhering to the constraints defined by Documentum CM Server.	integer	>=0	0
	modifier	Specifies the modifier of the contents of a SysObject for which checksum is requested. For the primary content, the modifier is ignored.  For example, the value of the modifier can be: modifier =<mod1> or <mod2>.	string	string	null

### 2.15.3.2.2 Request media types

Not applicable

### 2.15.3.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.15.3.2.4 Request body

Not applicable

### 2.15.3.2.5 Response headers

- Content-Digest

Contains the checksum of all the contents of an object as comma-separated values in the order of imported contents. If a criterion or criteria is specified in the query parameters, then the response header contains the checksum of an object based on the criterion or criteria chosen in the query parameters.

### 2.15.3.2.6 Response status

- 200 - Checksum retrieved successfully and provided as part of the Content-Digest header
- 404 - Checksum is not found for a requested object and results with an exception

### 2.15.3.2.7 Response body

Not applicable

## 2.16 Comment (s)

### 2.16.1 Comment

This resource allows you to work with comment objects.

### 2.16.1.1 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">"Comment" on page 151</a>
parent	The parent Comment	<a href="#">"Comment" on page 151</a>
replies [1]	The replies to the Comment	<a href="#">"Comment Replies" on page 163</a>
delete [1]	Delete the Comment	
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.16.1.2 Feed

No feed support.

### 2.16.1.3 Operations

#### 2.16.1.3.1 Supported HTTP methods

Method	Description
GET	Retrieves the comment
DELETE	Deletes the comment

#### 2.16.1.3.2 Get a comment

Retrieves a comment.

##### 2.16.1.3.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

**2.16.1.3.2.2 Request media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

**2.16.1.3.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.16.1.3.2.4 Request body**

N/A

**2.16.1.3.2.5 Response headers**

N/A

**2.16.1.3.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

**2.16.1.3.2.7 Response status**

- 200 - Comment retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.16.1.3.2.8 Response body**

 **Example 2-50: XML response**

```
<comment xmlns="http://identifiers.emc.com/vocab/documentum"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <object-id>0800000180143727</object-id>
  <comment-id>1324839</comment-id>
  <owner-name>John Doe</owner-name>
  <creation-date>2016-01-26T15:16:16.000+00:00</creation-date>
  <modified-date>2016-01-26T15:16:16.000+00:00</modified-date>
  <content-value>The spec looks good.&nbsp;
    Just two comments<br&gt;&lt;ol&gt;&lt;li&gt;Please add a diagram
    to show the
      flow.<br&gt;&lt;li&gt;Please clarify the filter
    criteria.&lt;br&gt;&lt;/li&gt;
  </content-value>
</comment>
```

```

        &lt;/ol&gt;
    </content-value>
<parent-id>0</parent-id>
<title/>
<can-delete>true</can-delete>
<can-reply>true</can-reply>
<links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/
        objects/0900000180143717/comments/1324839"/>
    <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/
REPO/
        objects/0900000180143717/comments"/>
    <link rel="http://identifiers.emc.com/linkrel/replies"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000180143717/comments/1324839/replies"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000180143717/comments/1324839"/>
</links>
</comment>

```



### Example 2-51: JSON response

```
{
    "object-id": "0800000180143727",
    "comment-id": "1324839",
    "owner-name": "John Doe",
    "creation-date": "2016-01-26T15:16:16.000+00:00",
    "modified-date": "2016-01-26T15:16:16.000+00:00",
    "content-value": "The spec looks good.&nbsp;Just two
comments<br&gt;<ol&gt;<li&gt;
        Please add a diagram to show the flow.&lt;/font&gt;&lt;/font&gt;&lt;/
font&gt;;
        &lt;/li&gt;&lt;li&gt;Please clarify the filter criteria.&lt;br&gt;&lt;/
li&gt;
        &lt;/ol&gt;",
    "parent-id": "0",
    "title": "",
    "can-delete": true,
    "can-reply": true,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000180143717/comments/1324839"
        },
        {
            "rel": "parent",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000180143717/comments"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/replies",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000180143717/comments/1324839/replies"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",

```

```
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324839"
    }
}
```



### 2.16.1.3.3 Delete a comment

This method allows the user who created the comment to delete the comment. The delete link for a comment is available only when a user can delete that comment.

#### 2.16.1.3.3.1 HTTP Request method and query parameters

DELETE

#### 2.16.1.3.3.2 Request media types

N/A

#### 2.16.1.3.3.3 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.16.1.3.3.4 Request body

N/A

#### 2.16.1.3.3.5 Response headers

N/A

#### 2.16.1.3.3.6 Response media types

N/A

#### 2.16.1.3.3.7 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.16.1.3.3.8 Response body

N/A

## 2.16.2 Comments

This resource allows you to work with a comment objects collection.

### 2.16.2.1 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Comments on object {objectId}
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: [“Comments” on page 156](#)
- Entry ID: Comment resource URI
- Entry Title: Comment's title
- Entry Summary: Comment {commentId} on object {objectId}
- Entry Updated: Comment's modification date
- Entry Published: Comment's creation date

### 2.16.2.2 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">“Comments” on page 156</a>

### 2.16.2.3 Operations

#### 2.16.2.3.1 Supported HTTP methods

Method	Description
GET	Retrieves a list of root-level comments
POST	Create a root-level comment

### 2.16.2.3.2 Get root-level comments

Retrieves the list of root level comments for the object.

#### 2.16.2.3.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- page
- items-per-page
- include-total
- filter
- links
- sort

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.16.2.3.2.2 Request media types

N/A

#### 2.16.2.3.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.16.2.3.2.4 Request body

N/A

#### 2.16.2.3.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.16.2.3.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.16.2.3.2.7 Response status

- 200 - Comments retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.16.2.3.2.8 Response body

##### Example 2-52: XML response

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
    comments</id>
  <title>Comments on object: 0900000180143717 under parent comment id: 0
  </title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2016-02-01T21:50:18.768+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/
documentum">100
  </dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
objects/
  09000001800d86ec/comments"/>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324839</id>
    <title></title>
    <author>
      <name>John Doe</name>
      <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/John
%2BDoe
      </uri>
      </author>
      <summary>&lt;font color= '#363636' &gt;&lt;font size= '2' &gt;
&lt;font face= 'Arial' &gt;The spec looks good.& Just two
comments&lt;br&gt;
      </summary>
    </entry>
  </feed>
```

```

        &lt;/font&gt;&lt;/font&gt;&lt;/font&gt;&lt;ol&gt;&lt;li&gt;
            &lt;font color= '#363636' &gt;&lt;font size= '2' &gt;
                &lt;font face= 'Arial' &gt;Please add a diagram to show the
            flow.&lt;/font&gt;
                &lt;/font&gt;&lt;/font&gt;&lt;li&gt;&lt;font color=
            '#363636' &gt;
                &lt;font size= '2' &gt;&lt;font face= 'Arial' &gt;Please clarify
            the filter
                criteria.&lt;br&gt;&lt;/font&gt;&lt;/font&gt;&lt;/font&gt;&lt;li&gt;
                    &lt;/ol&gt;</summary>
        <updated>2016-01-26T15:16:16.000+00:00</updated>
        <published>2016-01-26T15:16:16.000+00:00</published>
        <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/
            objects/0900000180143717/comments/1324839"/>
        <content type="application/xml"
            src="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000180143717/comments/1324839"/>
    </entry>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000180143717/comments/1324841</id>
        <title></title>
        <author>
            <name>Jane Doe</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
                Jane%2BDoe</uri>
        </author>
        &lt;summary&gt;&lt;font face= 'Arial' &gt;&lt;font size= '2' &gt;
            &lt;font color= '#363636' &gt;Should we also support paging?
        &lt;br&gt;
            &lt;/font&gt;&lt;/font&gt;&lt;/font&gt;&lt;/summary&gt;
        <updated>2016-01-26T15:19:57.000+00:00</updated>
        <published>2016-01-26T15:19:57.000+00:00</published>
        <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/
            objects/0900000180143717/comments/1324841"/>
        <content type="application/xml" src="http://localhost:8080/dctm-rest/
            repositories/REPO/objects/0900000180143717/comments/1324841"/>
    </entry>
</feed>

```



### ➡ Example 2-53: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000180143717/comments",
    "title": "Comments on object: 0900000180143717 under parent comment id: 0",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2016-02-01T22:07:00.401+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {

```

```
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments"
    }
],
"entries": [
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
    comments/1324839",
    "title": "",
    "author": [
        {
            "name": "John Doe",
            "uri": "http://localhost:8080/dctm-rest/repositories/REPO/users/John
%2BDoe"
        }
    ],
    "summary": "<font color= '#363636' >&lt;font size= '2'
&gt;&lt;font face= 'Arial' >
    The spec looks good. Just two comments<br&gt;&lt;/font&gt;&lt;/
font&gt;&lt;/font&gt;
        &lt;ol&gt;&lt;li&gt;&lt;font color= '#363636' >&lt;font
size= '2' &gt;
            &lt;font face= 'Arial' >Please add a diagram to show the
flow.&lt;/font&gt;
                &lt;/font&gt;&lt;/font&gt;&lt;/li&gt;&lt;li&gt;&lt;font color=
'#363636' >
                    &lt;font size= '2' &gt;
                        &lt;font face= 'Arial' >Please clarify the filter
criteria.&lt;br&gt;
                    &lt;/font&gt;&lt;/font&gt;&lt;/li&gt;&lt;/ol&gt;",
    "updated": "2016-01-26T15:16:16.000+00:00",
    "published": "2016-01-26T15:16:16.000+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324839"
        }
    ],
    "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324839"
    }
},
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
    comments/1324841",
    "title": "",
    "author": [
        {
            "name": "Jane Doe",
            "uri": "http://localhost:8080/dctm-rest/repositories/REPO/users/Jane
%2BDoe"
        }
    ],
    "summary": "<font face= 'Arial' >&lt;font size= '2' &gt;
        &lt;font color= '#363636' >Should we also support paging?
&lt;br&gt;&lt;/font&gt;
```

```

    &lt;/font&gt;&lt;/font&gt;" ,
"updated": "2016-01-26T15:19:57.000+00:00",
"published": "2016-01-26T15:19:57.000+00:00",
"links": [
{
    "rel": "edit",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324841"
}
],
"content":{
    "type": "application/vnd.emc.documentum+json",
    "src": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324841"
}
]
}

```



### 2.16.2.3.3 Create a root-level comment

You can use this method to add a root-level comment. You must have *<RELATE>* permission on the object to add a comment. The content value of a comment can be either plain text or rich text. All content is sanitized before it is saved to the repository.



**Note:** There is a limitation in the Collaboration Service that requires you to have *<WRITE>* permission on the object in order to start a comment thread. In other words, a user requires *<WRITE>* permission to create the first comment, but only requires *<RELATE>* permission to add subsequent comments.

#### 2.16.2.3.3.1 HTTP Request method and query parameters

POST

#### 2.16.2.3.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

### 2.16.2.3.3.3 Request headers

- Authorization
- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.16.2.3.3.4 Request body

The content-value tag contains the rich text content of the comment. It's required for comment creation.

### 2.16.2.3.3.5 Response headers

N/A

### 2.16.2.3.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

### 2.16.2.3.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.16.2.3.3.8 Response body

#### ➡ Example 2-54: XML response

```
<comment xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <content-value>
        The spec looks good. Just two comments:&lt;br&gt;
        &lt;ol&gt;
            &lt;li&gt;Please add a diagram to show the flow.&lt;/li&gt;
            &lt;li&gt;Please clarify the filter criteria.&lt;br&gt;&lt;/li&gt;
        &lt;/ol&gt;
    </content-value>
</comment>
```



 **Example 2-55: JSON response**

```
{
    "content-value": "The spec looks good. Just two comments:&lt;br&gt;
        &lt;ol&gt;&lt;li&gt;Please add a diagram to show the flow.&lt;/li&gt;
        &lt;li&gt;Please clarify the filter criteria.&lt;br&gt;&lt;/li&gt;
        &lt;/ol&gt;"
}
```



## 2.17 Comment Replies

This resource allows you to work with a list of comment replies.

### 2.17.1 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Comments on object {objectId} in reply to comment {parentCommentId}
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: A reply comment to the object
- Entry ID: Comment resource URI
- Entry Title: Comment's title
- Entry Summary: Comment {commentId} on object {objectId} in reply to comment {parentCommentId}
- Entry Published: Comment's creation date
- Entry Updated: Comment's modification date

### 2.17.2 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">"Comment Replies" on page 163</a>

## 2.17.3 Operations

### 2.17.3.1 Supported HTTP methods

Method	Description
GET	Retrieves a list of replies for a comment
POST	Create a reply for a comment

### 2.17.3.2 Get replies

Retrieves the list of replies to a comment.

#### 2.17.3.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- page
- items-per-page
- include-total
- filter
- links
- sort

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.17.3.2.2 Request media types

N/A

#### 2.17.3.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.17.3.2.4 Request body

N/A

#### 2.17.3.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.17.3.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.17.3.2.7 Response status

- 200 - Replies retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.17.3.2.8 Response body

##### Example 2-56: XML response

```
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
      comments/1324839/replies</id>
    <title>Comments on object: 0900000180143717 under parent comment id:
1324839</title>
    <author>
      <name>OpenText Documentum</name>
    </author>
    <updated>2016-02-12T19:47:24.307+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/
documentum">100
    </dm:items-per-page>
    <linkrel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
objects/
0900000180143717/
      comments/1324839/replies"/>
    <entry>
      <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
      comments/1324840</id>
      <title/>
      <author>
        <name>John Doe</name>
```

```

<uri>http://localhost:8080/dctm-rest/repositories/REPO/users/John
%2BDoe</uri>
</author>
<summary>
    &li;font face= 'Arial' &gt;
    &li;font size= '2' &gt;
    &li;font color= '#363636' &gt;Thanks for reviewing it.& I'll update the
spec.
    &li;br&gt;
    &li;/font&gt;&li;/font&gt;&li;/font&gt;
</summary>
<updated>2016-01-26T15:19:21.000+00:00</updated>
<published>2016-01-26T15:19:21.000+00:00</published>
<linkrel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/objects/
0900000180143717/comments/1324840"/>
<contenttype="application/xml" src="http://localhost:8080/dctm-rest/
repositories/
REPO/objects/0900000180143717/comments/1324840"/>
</entry>
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
        comments/1324842</id>
    <title/>
    <author>
        <name>Jane Doe</name>
        <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/Jane
%2BDoe</uri>
        </author>
        <summary>Do we need to expose this feature through REST api?
&li;br&gt;</summary>
        <updated>2016-01-26T15:27:50.000+00:00</updated>
        <published>2016-01-26T15:27:50.000+00:00</published>
        <linkrel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/objects/
0900000180143717/comments/1324842"/>
        <contenttype="application/xml" src="http://localhost:8080/dctm-rest/
repositories/
REPO/objects/0900000180143717/comments/1324842"/>
    </entry>
</feed>
```



### Example 2-57: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/comments/1324839/replies",
    "title": "Comments on object: 0900000180143717 under parent comment id:
1324839",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2016-02-12T19:29:04.092+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
```

```
{
  "rel": "self",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
  comments/1324839/replies"
}
],
"entries": [
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
  0900000180143717/comments/1324840",
  "title": "",
  "author": [
    {
      "name": "John Doe",
      "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
users/John%2BDoe"
    }
  ],
  "summary": "&lt;font face= 'Arial' &gt;&lt;font size= '2' &gt;
  &lt;font color= '#363636' &gt;
    Thanks for reviewing it. I'll update the spec.&lt;br&gt;
    &lt;/font&gt;&lt;/font&gt;&lt;/font&gt;",
  "updated": "2016-01-26T15:19:21.000+00:00",
  "published": "2016-01-26T15:19:21.000+00:00",
  "links": [
    {
      "rel": "edit",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
  objects/
  0900000180143717/comments/1324840"
    }
  ],
  "content": {
    "type": "application/vnd.emc.documentum+json",
    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
  objects/
  0900000180143717/comments/1324840"
  }
},
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
  0900000180143717/comments/1324842",
  "title": "",
  "author": [
    {
      "name": "Jane Doe",
      "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
users/Jane%2BDoe"
    }
  ],
  "summary": "Do we need to expose this feature through REST api?
&lt;br&gt;",
  "updated": "2016-01-26T15:27:50.000+00:00",
  "published": "2016-01-26T15:27:50.000+00:00",
  "links": [
    {"rel": "edit", "href": "http://localhost:8080/dctm-rest/
  repositories/
  REPO/objects/0900000180143717/comments/1324842"...
  ]
  ],
  "content": {
    "type": "application/vnd.emc.documentum+json",
    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
  objects/
  0900000180143717/comments/1324842"
  }
}
]
```

```
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
objects/
    }
}
]
```



### 2.17.3.3 Create a reply to a comment

You can use this method to add a reply to a comment. You must have <RELATE> permission on the comment object in order to add a reply. The content value of a comment reply can be either plain text or rich text. All content is sanitized before it is saved to the repository.

#### 2.17.3.3.1 HTTP Request method and query parameters

POST

This method supports the following common query parameters:

- `comment-id`: The id of the comment
- `parent-id`: The id of the parent comment
- `content-value`: Contains the text content of the comment
- `can-delete`: Indicates whether the user can delete this comment
- `can-reply`: Indicates whether the user can reply to this comment

#### 2.17.3.3.2 Request media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

#### 2.17.3.3.3 Request headers

- `Authorization`
- `Content-Type`

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.17.3.3.4 Request body

The content-value tag contains the encoded text content of the reply. It's required for comment creation.

##### ➡ Example 2-58: XML request

```
<comment xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <content-value>
        Agree with John.&lt;br&gt;&lt;ol&gt;&lt;li&gt;Please add a diagram to show
        the
        flow.&lt;/li&gt;&lt;li&gt;Please clarify the filter
        criteria.&lt;br&gt;&lt;/li&gt;
        &lt;/ol&gt;
    </content-value>
</comment>
```



##### ➡ Example 2-59: JSON request

```
{
    "content-value": "Agree with John.&lt;br&gt;&lt;ol&gt;&lt;li&gt;Please add a diagram to
    show the flow.&lt;/li&gt;&lt;li&gt;Please clarify the filter
    criteria.&lt;br&gt;&lt;/li&gt;&lt;/ol&gt;"
}
```



#### 2.17.3.3.5 Response headers

- Location

#### 2.17.3.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.17.3.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.17.3.3.8 Response body

 **Example 2-60: XML response**

```

<comment xmlns="http://identifiers.emc.com/vocab/documentum"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <object-id>0800000180143727</object-id>
  <comment-id>1324839</comment-id>
  <owner-name>John Doe</owner-name>
  <creation-date>2016-01-26T15:16:16.000+00:00</creation-date>
  <modified-date>2016-01-26T15:16:16.000+00:00</modified-date>
  <content-value>Agree with John.&lt;br&gt;&lt;ol&gt;&lt;li&gt;Please add a
diagram
      to show the flow.&lt;/li&gt;&lt;li&gt;Please clarify the filter
criteria.&lt;br&gt;
      &lt;/li&gt;&lt;/ol&gt;
  </content-value>
  <parent-id>1324830</parent-id>
  <title/>
  <can-delete>true</can-delete>
  <can-reply>true</can-reply>
  <links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/
      objects/0900000180143717/comments/1324839"/>
    <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/
REPO/
      objects/0900000180143717/comments"/>
    <link rel="http://identifiers.emc.com/linkrel/replies"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
      comments/1324839/replies"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
      comments/1324839"/>
  </links>
</comment>
```



 **Example 2-61: JSON response**

```
{
  "object-id": "0800000180143727",
  "comment-id": "1324839",
  "owner-name": "John Doe",
  "creation-date": "2016-01-26T15:16:16.000+00:00",
  "modified-date": "2016-01-26T15:16:16.000+00:00",
  "content-value": "Agree with John.&lt;br&gt;&lt;ol&gt;&lt;li&gt;Please add a
diagram to
      show the flow.&lt;/li&gt;&lt;li&gt;Please clarify the filter criteria.
      &lt;br&gt;&lt;/li&gt;&lt;/ol&gt;",
  "parent-id": "1324830",
  "title": "",
  "can-delete": true,
  "can-reply": true,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
```

```

0900000180143717/
    comments/1324839"
},
{
    "rel": "parent",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
    comments"
},
{
    "rel": "http://identifiers.emc.com/linkrel/replies",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
    comments/1324839/replies"
},
{
    "rel": "http://identifiers.emc.com/linkrel/delete",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000180143717/
    comments/1324839"
}
]
}

```



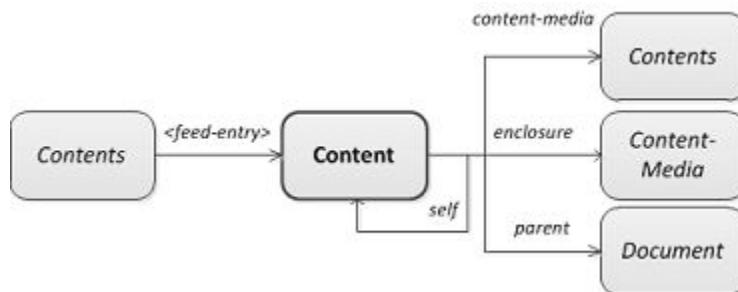
## 2.18 Content (s)

### 2.18.1 Content

The Content resource represents a content object in a repository. A content object contains both content properties and a link to the content stream.

#### 2.18.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also: “Contents” on page 178 and About the Diagram on page 13.*

### 2.18.1.2 Feed

No feed support.

### 2.18.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this content.	<a href="#">"Content" on page 171</a>
content-media [1] *	Link to the content binary stream for download.	Content URL from REST server when neither of the OpenText Documentum Content Management (CM) Accelerated Content Services server or OpenText Documentum Content Management (CM) Branch Office Caching Services server is available. ( <a href="#">"Content Media" on page 185</a> )  Distributed Content URL when either of the Accelerated Content Services server or Branch Office Caching Services server is available.
enclosure [1] *	Link to content binary stream for download.	Content URL from REST server when neither of the Accelerated Content Services server or Branch Office Caching Services server is available. ( <a href="#">"Content Media" on page 185</a> )  Distributed Content URL when either of the Accelerated Content Services server or Branch Office Caching Services server is available.
parent	Link to the associated object of this content.	<a href="#">"Document" on page 214</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		
* Repeatable link relation		

### 2.18.1.4 Operations

#### 2.18.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the metadata of the primary content or of the rendition content for an existing document object, optionally with content media links for content download.
DELETE	Deletes primary content and/or rendition content for a document object.

#### 2.18.1.4.2 Get content

Retrieve the metadata of the primary content or of the rendition content for an existing document object, optionally with content media links for content download.

#### 2.18.1.4.2.1 Content URL return policy

By using the `media-url-policy` parameter, you can customize content URL return policy in this operation to control what URL to put in the `content-media` and `enclosure` link relations. Valid values for this parameter are:

Value	Description
DC-PREF*	<p>When you set <code>media-url-policy</code> to <code>DC-PREF</code>, this operation tries to retrieve a Distributed Content URL first. If no Distributed Content URL is available, a Content Media resource URL from the REST server is still acceptable. Details are as follows:</p> <ul style="list-style-type: none"> <li>• If the Branch Office Caching Services server is available, and the network location is specified correctly, the Branch Office Caching Services content URL corresponding to the network location is returned.</li> <li>• If the Branch Office Caching Services server is available, but the specified network location is not correct, or there is no Branch Office Caching Services configured for this network location, or the Branch Office Caching Services server is unavailable, the Accelerated Content Services content URL is returned if any Accelerated Content Services server is available.</li> <li>• If there are multiple Accelerated Content Services URLs available, the REST server chooses the one with most proximity.</li> <li>• If no Accelerated Content Services or Branch Office Caching Services server is available, a Content Media resource URL is returned from the REST server.</li> </ul>
DC-ONLY	<p>When you set <code>media-url-policy</code> to <code>DC-ONLY</code>, the REST server returns a Distributed Content URL (either from the Accelerated Content Services server or the Branch Office Caching Services server with the same logic as the <code>DC-PREF</code> option does. If there is no Accelerated Content Services server or Branch Office Caching Services server, the REST server returns the HTTP 400 Conflict error.</p>

Value	Description
LOCAL	When you set <code>media-url-policy</code> to LOCAL, the REST server returns a Content Media resource URL from the REST server regardless of the availability of the Accelerated Content Services server and the Branch Office Caching Services server.
ALL	When you set <code>media-url-policy</code> to ALL, all available Content Media links are returned, each one pointing to a different content media server. It includes all available Accelerated Content Services URLs, all available Branch Office Caching Services URLs, and the local Content Media resource URL.
* Default	

#### 2.18.1.4.2.2 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page	Specifies the page number for the content imported.	integer	Any non-negative number. Documentum CM Server defines the other constraints on this value.	0
	format	Specifies the format for the content imported.	string	Any format registered in the Format table of the repository.	null
	modifier	Specifies the page modifier for the content.	string	Any string value.	null
	media-url-policy	Specifies the Content Media URL return policy. For detailed information, see “Content URL return policy” on page 173.	enum	<ul style="list-style-type: none"> <li>• DC-PREF</li> <li>• DC-ONLY</li> <li>• LOCAL</li> <li>• ALL</li> </ul>	DC-PREF

HTTP method	Query parameters	Description	Data type	Value range	Default value
	network-location	Specifies the network location identifier for Branch Office Caching Services content download.  Used only when media-url-policy is set to DC-PREF or DC-ONLY.	string	Any valid network location identifier corresponding to the netloc_ident property of the dm_network_location_map instance.	null

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.18.1.4.2.3 Request media types

N/A

#### 2.18.1.4.2.4 Request headers

- Accept
- Authorization

#### 2.18.1.4.2.5 Request body

N/A

#### 2.18.1.4.2.6 Response headers

- Content-Type

#### 2.18.1.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.18.1.4.2.8 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.18.1.4.2.9 Response body

XML or JSON message that represents the content metadata and a URL for content download.

#### 2.18.1.4.3 Delete content

Delete primary content and/or rendition content for a document object. By default, when a primary content is deleted, the renditions on the same page are deleted at the same time. Renditions created by Foundation REST API are always deleted if the corresponding primary content is deleted. However, under some circumstances, renditions created using other approaches can remain when the primary content is deleted.

#### 2.18.1.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	page	Specifies the page number for the content imported.	integer	Any non-negative number. Documentum CM Server defines the other constraints on this value.	NA This is a required parameter.
	format	Specifies the format for the content imported.	string	Any format registered in the Format table of the repository.	NA This is a required parameter.
	modifier	Specifies the page modifier for the content.	string	Any string value.	null

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

**2.18.1.4.3.2 Request media types**

N/A

**2.18.1.4.3.3 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.18.1.4.3.4 Request body**

N/A

**2.18.1.4.3.5 Response headers**

N/A

**2.18.1.4.3.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

**2.18.1.4.3.7 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.18.1.4.3.8 Response body**

HTTP 204 No Content status upon a successful delete operation. The response body contains no content.

## 2.18.2 Contents

The **Contents** resource represents a collection of primary content objects and renditions of an object. Each content entry contains content properties and a link to the content binary stream.

### 2.18.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

### 2.18.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the content without the file extension.
- Feed Title: Object name.
- Feed Updated: Last updated time of the object.
- Supports POST: Yes
- Entry: “[Content](#)” on page 171
- Entry ID: URI of the content
- Entry Title: Content name
- Entry Summary: Content description
- Entry Updated: `r_modify_date` of the content

### 2.18.2.3 Link relations

Link relation	Description	Resource reference
self	Link to the contents.	“ <a href="#">Contents</a> ” on page 178
parent [1]	Link to the associated object or document of this content	“ <a href="#">Document</a> ” on page 214
first, last, next, previous	Pagination links.	“ <a href="#">Folder Child Objects</a> ” on page 282
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

## 2.18.2.4 Operations

### 2.18.2.4.1 Supported HTTP methods

Method	Description
GET	GET a collection of content resources.
POST	Create a new primary content or rendition on a document.

### 2.18.2.4.2 Get content

GET a collection of all primary content, rendition content metadata, and URLs to the content binary stream for a document object, depending on the query parameter settings.

#### 2.18.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	media-url-policy	Specifies the Content Media URL return policy. For detailed information, see “Content URL return policy” on page 173.	enum	<ul style="list-style-type: none"> <li>• DC-PREF</li> <li>• DC-ONLY</li> <li>• LOCAL</li> <li>• ALL</li> </ul>	null
	network-location	Specifies the network location identifier for Branch Office Caching Services content download. Used only when media-url-policy is set to DC-PREF or DC-ONLY.	string	Any valid network location identifier corresponding to the netloc_ident property of the dm_network_location_map instance.	null

This operation also supports the following parameters:

- inline
- page

- items-per-page
- include-total

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### **2.18.2.4.2.2 Request media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### **2.18.2.4.2.3 Request headers**

- Accept
- Authorization

#### **2.18.2.4.2.4 Request body**

N/A

#### **2.18.2.4.2.5 Response headers**

- Content-Type

#### **2.18.2.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing)

#### **2.18.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

#### 2.18.2.4.2.8 Response body

XML or JSON message that represents the content, and a link to the content binary stream.

#### 2.18.2.4.3 Create content

Create new primary content or renditions on a document.

#### 2.18.2.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	primary	Specifies whether the imported content is primary content or a rendition.	boolean	<p><i>true</i> - indicates the imported content is primary content.</p> <p><i>false</i> - indicates the imported content is a rendition.</p>	<i>true</i>
	page	Specifies the page number for the content imported.	integer	<p>Any non-negative number.</p> <p>Documentum CM Server defines the other constraints on this value.</p>	0
	format	Specifies the format for the content imported.	string	Any format registered in the Format table of the repository.	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	overwrite	Specifies whether or not to overwrite the existing content with the same content properties.	string	<i>true</i> - Overwrite the existing primary content. <i>false</i> - Throw an exception when the primary content or rendition with the same content properties already exists.	<i>false</i>
	modifier	Specifies the page modifier for the content.	string	Any string value.	null
	content-length	Specifies the byte count of the contents to be uploaded	string	An accurate byte count of the content to be uploaded	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-charset	<p>Used to sanitize content.</p> <p>This parameter tells the server how to parse the content for sanitizing. When the content-charset is not provided, the server tries to get metadata information from the content itself when it's in HTML format.</p> <p>When the metadata has a valid charset, then it is used to sanitize the content. The configuration default charset is used when the content-charset parameter and the metadata information are not found.</p> <p> <b>Note:</b> When the charset and the content do not</p>	string	A valid charset	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
		match, the uploaded content may have an incorrectly encoded value.			



### Notes

- When you import primary content, the `format` parameter is optional. However, when you specify this parameter, its value must be the same as the `format` of the primary content on page 0. When the page contains primary content, the `overwrite` parameter must be set to `true`.
- When storing content to Centera storage, the `content-length` query parameter must be provided. When providing the value for `content-length`, you must provide an accurate byte count of the content that you want to upload.

For other storage mediums, such as harddisk, the `content-length` parameter is not required and is optional. However, even when not required, when this parameter is provided, it must be accurate.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.18.2.4.3.2 Request media types

Any MIME type registered in the Format table of the repository.

#### 2.18.2.4.3.3 Request headers

- `Accept`
- `Authorization`
- `Content-Type`

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.18.2.4.3.4 Request body

Binary of the content.

#### 2.18.2.4.3.5 Response headers

- Location
- Content-Type

#### 2.18.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.18.2.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.18.2.4.3.8 Response body

JSON or XML representation of the Content resource.

## 2.19 Content Media

The Content Media resource represents the media of the primary content or rendition downloaded from the REST server, other than those from an Accelerated Content Services server or a Branch Office Caching Services server. This resource is typically used when no Accelerated Content Services server or Branch Office Caching Services server is available.

### 2.19.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also: “Content” on page 171 and [About the Diagram on page 13](#).*

## 2.19.2 Feed

No feed support.

## 2.19.3 Link relations

N/A

## 2.19.4 Operations

### 2.19.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a content media from the REST server.

### 2.19.4.2 Get content media from the REST server

Retrieve a content media from the REST server when no Accelerated Content Services server or Branch Office Caching Services server is available.

#### 2.19.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page	Specifies the page number for the content imported.	integer	Any non-negative number. Documentum CM Server defines the other constraints on this value.	0
	format	Specifies the format for the content imported.	string	Any format registered in the Format table of the repository.	null
	modifier	Specifies the page modifier for the Content Media.	string	Any string value.	null

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.19.4.2.2 Request media types

N/A

#### 2.19.4.2.3 Request headers

- Authorization
- If-None-Match: If the REST client uses the ETag mechanism to enable client side web cache, the value of this header must be identical to the ETag value from the previous server response header.



#### Notes

- When the client provides the If-None-Match header, the REST server compares the value of the header with the ETag of the current version of the resource. If these two values match, meaning that the resource has not been changed, the REST server sends back a short response with the HTTP 304 Not Modified status, which notifies the client to use the cached version.
- REST APIs such as the GET operation on content and contentMedia uses or returns only content-media.

#### 2.19.4.2.4 Request body

N/A

#### 2.19.4.2.5 Response headers

- Content-Type
- Content-Length
- ETag

#### 2.19.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.19.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.19.4.2.8 Response body

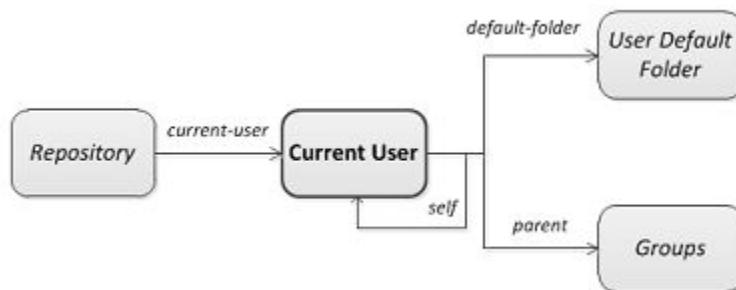
The content media downloaded from the REST server.

## 2.20 Current User

The Current User resource represents the metadata of the user that is currently logged in to the repository.

### 2.20.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

### 2.20.2 Feed

No feed support.

### 2.20.3 Link relations

Link relation	Description	Resource reference
self	This user resource.	<a href="#">"Current User" on page 188</a>
default-folder [1]	Default folder (home cabinet) of the user.	<a href="#">"User Default Folder" on page 780</a>
parent	Groups to which the user belongs.	<a href="#">"Groups" on page 317</a>
edit	This user resource. Only a SysAdmin or SuperUser can find this link.	<a href="#">"Current User" on page 188</a>
delete	This user resource. Only a SysAdmin or SuperUser can find this link.	<a href="#">"Current User" on page 188</a>
permission-set	The permission set definition of the user.	<a href="#">"User Permission Set" on page 784</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
`http://identifiers.emc.com/linkrel/`.

### 2.20.4 Operations

#### 2.20.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the metadata of the user that is currently logged in to the repository.

#### 2.20.4.2 Get a current user

Retrieve the metadata of a current user.

##### 2.20.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameter:

- view

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### **2.20.4.2.2 Request media types**

N/A

#### **2.20.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.20.4.2.4 Request body**

N/A

#### **2.20.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.20.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.20.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.20.4.2.8 Response body**

XML or JSON representation of the current login user.

## 2.21 Current User Preference (s)

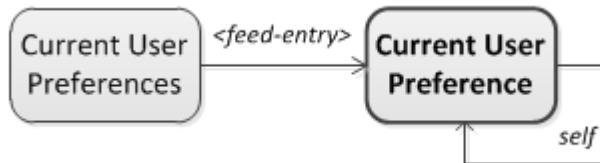
### 2.21.1 Current User Preference

The Current User Preference resource represents the user preference settings of the current user. This resource provides the user that is currently logged in to the repository with the capability to manage the user preference setting. For example, you can use this resource to store client-related metadata.

The syntax and semantics of the user preference is determined by the client application that uses REST services to store preference data. Foundation REST API introduces a preference resource model to store user preference content as plain text content. In addition to the user preference content, there are several properties to describe the preference metadata. User preference content characters must be escaped by the REST client.

#### 2.21.1.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “[Current User Preferences](#)” on page 198 and [About the Diagram](#) on page 13.

#### 2.21.1.2 Feed

No feed support.

#### 2.21.1.3 Link relations

The following table lists link relations for the Current User Preference resource:

Link relation	Description	Resource reference
self	This User Preference resource.	<a href="#">“Current User Preference” on page 191</a>
edit	Edit this User Preference resource.	<a href="#">“Current User Preference” on page 191</a>
delete [1]	Delete this User Preference resource.	<a href="#">“Current User Preference” on page 191</a>

Link relation	Description	Resource reference
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.21.1.4 Operations

### 2.21.1.4.1 Supported HTTP methods

Method	Description
GET	GET metadata, content, and other information from the preference instance
POST	Updates the attributes of the preference instance
DELETE	Deletes the preference instance

### 2.21.1.4.2 Get the current user preference settings

Gets the attributes, content, and other information of the current user preference instance.

#### 2.21.1.4.2.1 HTTP Request method and query parameters

GET

#### 2.21.1.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.21.1.4.2.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.21.1.4.2.4 Request body**

N/A

**2.21.1.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.21.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.21.1.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.21.1.4.2.8 Response body**

 **Example 2-62: XML response**

```
<preference xmlns="http://identifiers.emc.com/vocab/documentum">
  <client>kw</client>
  <owner-name>dmadmin</owner-name>
  <title>user preference for knowledge worker</title>
  <subject>user preference</subject>
  <keywords>
    <item>knowledge worker</item>
    <item>cn</item>
  </keywords>
  <creation-date>2015-11-10T05:40:40.000+00:00</creation-date>
  <r_modify_date>2015-11-10T05:49:52.000+00:00</r_modify_date>
  <preference-content>
    &lt;user-locale&gt;
      cn
    &lt;/user-locale&gt;
    &lt;timezone&gt;
      gmt+8
    &lt;/timezone&gt;
  </preference-content>
  <links>
    <link rel="self" href="/repositories/REPO/currentuser-preferences/kw"/>
    <link rel="edit" href="/repositories/REPO/currentuser-preferences/kw"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
          href="/repositories/REPO/currentuser-preferences/kw"/>
  </links>
</preference>
```

```
</links>
</preference>
```



### Example 2-63: JSON response

```
{
  "name": "preference",
  "client": "kw",
  "owner-name": "dmadmin",
  "title": "user preference for knowledge worker",
  "subject": "user preference",
  "keywords": ["knowledge worker", "cn"],
  "creation-date": "2015-11-10T05:40:40.000+00:00",
  "r_modify_date": "2015-11-10T05:49:52.000+00:00",
  "preference-content": "{\"user-locale\":\"cn\", \"timezone\":\"gmt+8\"}"
  "links": [
    {
      "rel": "self",
      "href": "/repositories/REPO/currentuser-preferences/kw"
    },
    {
      "rel": "edit",
      "href": "/repositories/REPO/currentuser-preferences/kw"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/delete",
      "href": "/repositories/REPO/currentuser-preferences/kw"
    }
  ]
}
```



#### 2.21.1.4.3 Update user preference settings

Update the attributes for the preference instance. When the client provides undefined attributes in the request body, the server throws an exception.

##### 2.21.1.4.3.1 HTTP Request method and query parameters

POST

##### 2.21.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.21.1.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.21.1.4.3.4 Request body

XML or JSON representation of preference settings of the current user. You can update the following properties:

- title
- subject
- keywords
- preference-content

The `title`, `subject`, and `keywords` properties have the same property definitions as those in the `dm_SysObject` type. There is no size limit on `preference-content`. However, the actual size affects the REST request and response payload.

You can place a CDATA block or escaped string in the `preference-content` property to store client-related metadata as shown in the following samples:

##### ➡ Example 2-64: XML representation

```
<preference>
  <properties>
    <title>Chrome Browser</title>
    <subject>The browser of Chrome</subject>
    <keywords>
      <item>Firefox</item>
      <item>Mozilla</item>
    </keywords>
    <preference-content><![CDATA[<user_name>Michael</user_name>]]></preference-content>
  </properties>
</preference>
```



##### ➡ Example 2-65: JSON representation

```
{
  "properties": {
    "title": "Chrome Browser",
    "subject": "The browser of Chrome",
    "keywords": ["Firefox", "Mozilla"],
    "preference-content": "{$user_name| Michael}"
  }
}
```

```
    }  
}
```



**Note:** The content in the preference-content element is treated as a plain text, and it is not parsed, it is only stored and retrieved as a string.

#### 2.21.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.21.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.21.1.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.21.1.4.3.8 Response body

XML or JSON representation of the updated user preference settings.



##### Example 2-66: XML response

```
<?xml version='1.0' encoding='UTF-8' ?>  
<preference>  
  <properties>  
    <client>client2</client>  
    <owner_name>Administrator</owner_name>  
    <title>the title of a preference</title>  
    <subject>summary of this preference</subject>  
    <keywords>  
      <item>key1</item>  
      <item>key2</item>  
    </keywords>  
    <creation-date>2015-09-07T05:47:00.000+00:00</creation-date>  
    <r_modify_date>2015-09-07T06:06:39.000+00:00</r_modify_date>  
    <preference-content>  
      &lt;user_name&gt;Michael&lt;/user_name&gt;  
    </preference-content>  
  </properties>
```

```

<links>
  <link rel="self" href="localhost:8080/dctm-rest/repositories/REPO/
  currentuser-preferences/client2.xml" />
  <link rel="edit" href="localhost:8080/dctm-rest/repositories/REPO/
  currentuser-preferences/client2.xml" />
  <link rel="http://identifiers.emc.com/linkrel/delete"
  href="localhost:8080/dctm-rest/repositories/REPO/
  currentuser-preferences/client2.xml" />
</links>
</preference>

```



#### Example 2-67: JSON response

```

{
  properties: {
    client: "client2",
    owner_name: "Administrator",
    title: "the title of a preference",
    subject: "summary of this preference",
    keywords: ["key1", "key2"],
    creation_date: "2015-09-07T05:47:00.000+00:00",
    r_modify_date: "2015-09-07T06:06:39.000+00:00",
    preference-content: "{\"user_name\":\"Michael\"}",
  }
  links:
  {
    {
      rel: "self"
      href: "localhost:8080/dctm-rest/repositories/REPO/
      currentuser-preferences/client2"
    },
    {
      rel: "edit"
      href: "localhost:8080/dctm-rest/repositories/REPO/
      currentuser-preferences/client2"
    },
    {
      rel: "http://identifiers.emc.com/linkrel/delete"
      href: "localhost:8080/dctm-rest/repositories/REPO/
      currentuser-preferences/client2"
    };
  }
}

```



##### 2.21.1.4.4 Delete user preference settings

Delete user preference settings.

**2.21.1.4.4.1 HTTP Request method and query parameters**

DELETE

**2.21.1.4.4.2 Request media types**

N/A

**2.21.1.4.4.3 Request headers**

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.21.1.4.4.4 Request body**

N/A

**2.21.1.4.4.5 Response headers**

N/A

**2.21.1.4.4.6 Response media types**

N/A

**2.21.1.4.4.7 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.21.1.4.4.8 Response body**

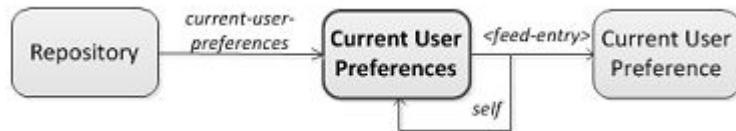
HTTP 204 No Content status upon a successful delete operation.

## **2.21.2 Current User Preferences**

The Preferences resource represents the collection of preferences metadata.

### 2.21.2.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “Current User” on page 188 and About the Diagram on page 13.

### 2.21.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Preferences
- Feed Updated: Server's current time
- Entry: “Current User Preference” on page 191
- Supports POST? Yes

### 2.21.2.3 Link relations

The following table lists link relations for the Preferences resource.

Link relation	Description	Resource reference
self	This object resource	“Current User Preferences” on page 198
first, last, next, previous	Pagination link	“Current User Preferences” on page 198

### 2.21.2.4 Operations

#### 2.21.2.4.1 Supported HTTP methods

Method	Description
GET	Gets the current user's preference collection.
POST	Creates a user preference for the current user.

#### **2.21.2.4.2 Get current user preferences collection**

GET the current user's preferences collection.

##### **2.21.2.4.2.1 HTTP Request method and query parameters**

GET

This method supports the following common query parameters:

- sort
- view
- inline
- page
- items-per-page
- include-total
- links
- filter

##### **2.21.2.4.2.2 Request media types**

N/A

##### **2.21.2.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### **2.21.2.4.2.4 Request body**

N/A

##### **2.21.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.21.2.4.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.21.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.21.2.4.2.8 Response body

##### Example 2-68: XML response

```
<?xml version='1.0' encoding='UTF-8' ?>
<feed>
  <id>http://localhost:8080/dctm-rest/repositories/REPO/currentuser/
preferences
  </id>
  <title>Preferences</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2015-08-17T04:36:37.195+00:00</updated>
  <dm:page>1</dm:page>
  <dm:items-per-page>100</dm:items-per-page>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/currentuser/
preferences.xml?inline=true" />
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/currentuser/
preferences/clinet1
    </id>
    <title>the title of a preference</title>
    <summary>summary of this preference</summary>
    <updated>2015-08-17T04:36:37.197+00:00</updated>
    <published>2015-08-17T04:36:37.197+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/
currentuser/preferences/clinet1.xml" />
    <content>
      <dm:preference xmlns:dm="http://identifiers.emc.com/vocab/
documentum">
        <dm:client>kw</dm:client>
        <dm:owner-name>dmadmin</dm:owner-name>
        <dm:title>user preference for knowledge worker</dm:title>
        <dm:subject>user preference</dm:subject>
        <dm:keywords>
          <dm:item>knowledge worker</dm:item>
          <dm:item>cn</dm:item>
        </dm:keywords>
        <dm:creation-date>2015-11-10T05:40:40.000+00:00
      </dm:preference>
    </content>
  </entry>
</feed>
```

```

        </dm:creation-date>
        <dm:r_modify_date>2015-11-10T05:49:52.000+00:00</
dm:r_modify_date>
        <dm:preference-content>
            &lt;user-locale&gt;cn&lt;/user-locale&gt;
            &lt;timezone&gt;gmt+8&lt;/timezone&gt;
        </dm:preference-content>
        <links>
            <link rel="self" href="/repositories/REPO/
currentuser-preferences/kw"/>
            <link rel="edit" href="/repositories/REPO/
currentuser-preferences/kw"/>
            <link rel="http://identifiers.emc.com/linkrel/delete"
href="/repositories/REPO/currentuser-preferences/kw"/>
        </links>
    </dm:preference>
</content>
</entry>
</feed>

```



### Example 2-69: JSON response

```
{
    id: "http://localhost:8080/dctm-rest/repositories/REPO/currentuser/
preferences",
    title: "Preferences",
    author: [{name: "OpenText Documentum"}],
    updated: "2015-08-17T04:40:48.970+00:00",
    page: 1,
    items-per-page: 100,
    links:[{rel: "self",href: "http://localhost:8080/dctm-rest/repositories/
REPO/
    currentuser/preferences.json?inline=true"}],
    entries:[{
        id: "http://localhost:8080/dctm-rest/repositories/REPO/currentuser/
preferences/
        client1",
        title: "the title of a preference",
        summary: "summary of this preference",
        updated: "2015-08-17T04:40:48.972+00:00",
        published: "2015-08-17T04:40:48.972+00:00",
        content: {
            {
                "name": "preference",
                "client": "kw",
                "owner-name": "dmadmin",
                "title": "user preference for knowledge worker",
                "subject": "user preference",
                "keywords": ["knowledge worker", "cn"],
                "creation-date": "2015-11-10T05:40:40.000+00:00",
                "r_modify_date": "2015-11-10T05:49:52.000+00:00",
                "preference-content": "{\\"user-locale\\":\\"cn\\", \\"timezone\\":
                \"gmt+8\\\"}",
                "links": [
                    {
                        "rel": "self",
                        "href": "/repositories/REPO/currentuser-preferences/kw"
                    },
                    {

```

```

        "rel": "edit",
        "href": "/repositories/REPO/currentuser-preferences/kw"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "/repositories/REPO/currentuser-preferences/kw"
    }
]
}
}
}

```



#### 2.21.2.4.3 Create a user preference

Create a user preference for the current user.

##### 2.21.2.4.3.1 HTTP Request method and query parameters

POST

##### 2.21.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.21.2.4.3.3 Request headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.21.2.4.3.4 Request body

###### ➡ Example 2-70: XML request

```

<preference xmlns="http://identifiers.emc.com/vocab/documentum">
    <client>kw</client>
    <title>user preference for knowledge worker</title>
    <subject>user preference</subject>
    <keywords>
        <item>knowledge worker</item>
        <item>cn</item>
    </keywords>
    <preference-content>
        &lt;user-locale&gt;cn&lt;/user-locale&gt;
        &lt;timezone&gt;gmt+8&lt;/timezone&gt;
    </preference-content>
</preference>

```



▶ **Example 2-71: JSON request**

```
{  
    "client": "kw",  
    "title": "user preference for knowledge worker",  
    "subject": "user preference",  
    "keywords": ["knowledge worker", "cn"],  
    "preference-content": "&lt;user-locale&gt;cn&lt;/user-locale  
                           &gt;&lt;timezone&gt;gmt+8&lt;/timezone&gt;  
"  
}
```



#### 2.21.2.4.3.5 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.21.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.21.2.4.3.7 Response status

- 201 - Added successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.21.2.4.3.8 Response body

▶ **Example 2-72: XML response**

```
<preference xmlns="http://identifiers.emc.com/vocab/documentum">  
    <client>kw</client>  
    <owner-name>dmadmin</owner-name>  
    <title>user preference for knowledge worker</title>  
    <subject>user preference</subject>  
    <keywords>  
        <item>knowledge worker</item>  
        <item>cn</item>  
    </keywords>  
    <creation-date>2015-11-10T05:40:40.000+00:00</creation-date>  
    <r_modify_date>2015-11-10T05:49:52.000+00:00</r_modify_date>  
    <preference-content>  
        &lt;user-locale&gt;cn&lt;/user-locale&gt;  
        &lt;timezone&gt;gmt+8&lt;/timezone&gt;  
    </preference-content>  
    <links>
```

```

        <link rel="self" href="/repositories/REPO/currentuser-preferences/kw"/>
        <link rel="edit" href="/repositories/REPO/currentuser-preferences/kw"/>
        <link rel="http://identifiers.emc.com/linkrel/delete"
              href="/repositories/REPO/currentuser-preferences/kw"/>
    </links>
</preference>

```



### Example 2-73: JSON response

```
{
    "client": "kw",
    "owner-name": "dmadmin",
    "title": "user preference for knowledge worker",
    "subject": "user preference",
    "keywords": ["knowledge worker", "cn"],
    "creation-date": "2015-11-10T05:40:40.000+00:00",
    "r_modify_date": "2015-11-10T05:49:52.000+00:00",
    "preference-content": "&lt;user-locale&gt;cn&lt;/user-locale&gt;
                           &lt;timezone&gt;gmt+8&lt;/timezone&gt;",
    "links": [
        {
            "rel": "self",
            "href": "/repositories/REPO/currentuser-preferences/kw"
        },
        {
            "rel": "edit",
            "href": "/repositories/REPO/currentuser-preferences/kw"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "/repositories/REPO/currentuser-preferences/kw"
        }
    ]
}
```



## 2.22 Current Version

The Current Version resource represents the current version of the version tree of an object. A version tree has only one CURRENT version.

### 2.22.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

## 2.22.2 Feed

No feed support.

## 2.22.3 Link relations

Link relation	Description	Resource reference
canonica [1]	Canonical URI of the current version of the object	<a href="#">“SysObject” on page 692</a> <a href="#">“Document” on page 214</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

## 2.22.4 Operations

### 2.22.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the current version of an object.

#### 2.22.4.2 Get the current version

Retrieves the current version of a specified object from its version tree.

##### 2.22.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameter:

- view

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.22.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.22.4.2.3 Request headers

- Accept
- Authorization

#### 2.22.4.2.4 Request body

N/A

#### 2.22.4.2.5 Response headers

- Content-Type

#### 2.22.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.22.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.22.4.2.8 Response body

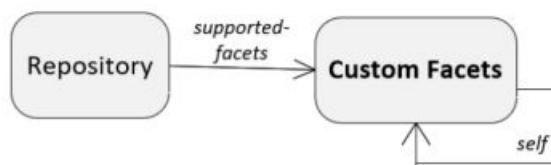
XML or JSON representation of the current version of the specified object.

## 2.23 Custom Facets

The *Custom Facets* resource allows you to retrieve the facet information that has been configured in xPlore. This facet information can then be used to filter search results.

### 2.23.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

## 2.23.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
- Feed Title: Supported Facets
- Feed Updated: Server's current time
- Supports POST: No
- Entry: A supported Facet object
- Entry ID: Request URI
- Entry Title: Facet name
- Entry Updated: Server's current time

## 2.23.3 Link relations

Link relation	Description	Resource reference
self	Link to itself.	<a href="#">“Custom Facets” on page 207</a>
first, last, next, previous	pagination links	<a href="#">“Custom Facets” on page 207</a>

## 2.23.4 Operations

### 2.23.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the Facets information that has been configured in xPlore.

### 2.23.4.2 Get supported Facets configured in xPlore

Retrieves the supported Facets that have been configured in xPlore.

#### 2.23.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, include-total	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.23.4.2.2 Request media types

N/A

#### 2.23.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.23.4.2.4 Request body

N/A

#### 2.23.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.23.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json
- application/hal+json

#### 2.23.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.23.4.2.8 Response body

 **Example 2-74: ATOM + XML Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REP01/supported-facets</id>
  <title>Supported Facets</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2018-11-19T06:23:49.941+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
  </dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REP01/
    supported-facets"/>
  <entry id="http://localhost:8080/dctm-rest/repositories/REP01/supported-facets</id>">
    <title>a_content_type</title>
    <type>string</type>
    <summary>Attribute a_content_type of Type string supports Faceting</summary>
    <updated>2018-11-19T06:23:49.941+00:00</updated>
    <published>2018-11-19T06:23:49.941+00:00</published>
    <link rel="search" href="http://localhost:8080/dctm-rest/repositories/REP01/
      search?facet=a_content_type&object-type=string&collections,
      include-total,inline,items-per-page,locations,page,q,
      sort,timezone,view"/>
  </entry>
  ...
</feed>
```



 **Example 2-75: JSON Response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REP01/supported-facets",
  "title": "Supported Facets",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2018-11-19T06:21:19.435+00:00",
```

```

    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
                     supported-facets"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO1/
                     supported-facets",
            "title": "a_content_type",
            "type": "string",
            "summary": "Attribute a_content_type of Type string supports
Faceting",
            "updated": "2018-11-19T06:21:19.453+00:00",
            "published": "2018-11-19T06:21:19.453+00:00",
            "links": [
                {
                    "rel": "search",
                    " hreftemplate": "http://localhost:8080/dctm-rest/repositories/
REPO1/search?facet=a_content_type&
object-type=string{&collections,include-total,
inline,items-per-page,locations,page,q,sort,
timezone,view}"
                }
            ]
        }
    ]
}

```



#### ➡ Example 2-76: HAL + JSON Response

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO1/supported-facets",
    "title": "Supported Facets",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-11-19T06:25:00.143+00:00",
    "page": 1,
    "items-per-page": 100,
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO1/
supported-facets",
                "title": "a_content_type",
                "type": "string",
                "summary": "Attribute a_content_type of Type string supports
Faceting",
                "updated": "2018-11-19T06:25:00.144+00:00",
                "published": "2018-11-19T06:25:00.144+00:00",
                "_links": {
                    "search": {
                        " hreftemplate": "http://localhost:8080/dctm-rest/
repositories/REPO1/search?"

```

```
        facet=a_content_type&
object-type=string{&collections,
include-total,inline,items-per-page,
locations,page,q,sort,timezone,view}"
    }
},
...
],
{
"_links": {
"self": {
"href": "http://localhost:8080/dctm-rest/repositories/REP01/
supported-facets"
}
}
}
```



#### 2.23.4.3 More Code Samples

##### ➡ Example 2-77: Request

```
GET /dctm-rest/repositories/REP01/supported-facets?page=2&items-per-page=2
&include-total=true HTTP/1.1
Host: localhost:8080
Authorization: Basic QWRtaW5pc3RyYXRvcjpQYXNzd29yZEAxMjM=
Accept:application/vnd.emc.documentum+json
```



##### ➡ Example 2-78: Response

```
Status 200 OK

{
  "id": "http://localhost:8080/dctm-rest/repositories/REP01/supported-facets",
  "title": "Supported Facets",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2018-11-20T08:16:03.039+00:00",
  "page": 2,
  "items-per-page": 2,
  "total": 10,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REP01/
supported-facets?page=2&items-per-page=2&include-total=true"
    },
    {
      "rel": "previous",
      "href": "http://localhost:8080/dctm-rest/repositories/REP01/
supported-facets?page=1&items-per-page=2&include-total=true"
    },
    {
      "rel": "next",
      "href": "http://localhost:8080/dctm-rest/repositories/REP01/
supported-facets?page=3&items-per-page=2&include-total=true"
    }
  ]
}
```

```
{
  "rel": "next",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
    supported-facets?page=3&items-per-page=2&include-total=true"
},
{
  "rel": "first",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
    supported-facets?page=1&items-per-page=2&include-total=true"
},
{
  "rel": "last",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
    supported-facets?page=5&items-per-page=2&include-total=true"
}
],
"entries": [
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO1/
    supported-facets?page=2&items-per-page=2&include-total=true",
  "title": "keywords",
  "type": "string",
  "summary": "Attribute keywords of Type string supports Faceting",
  "updated": "2018-11-20T08:16:03.040+00:00",
  "published": "2018-11-20T08:16:03.040+00:00",
  "links": [
    {
      "rel": "search",
      "hreftemplate": "http://localhost:8080/dctm-rest/repositories/
        REPO1/search?facet=keywords&
        object-type=string{&collections,include-total,
        inline,items-per-page,locations,page,q,sort,
        timezone,view}"
    }
  ]
},
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO1/
    supported-facets?page=2&items-per-page=2&include-total=true",
  "title": "r_modify_date",
  "type": "datetime",
  "summary": "Attribute r_modify_date of Type datetime supports
Faceting",
  "updated": "2018-11-20T08:16:03.041+00:00",
  "published": "2018-11-20T08:16:03.041+00:00",
  "links": [
    {
      "rel": "search",
      "hreftemplate": "http://localhost:8080/dctm-rest/repositories/
        REPO1/search?facet=r_modify_date&
        object-type=datetime{&collections,
        include-total,inline,items-per-page,locations,
        page,q,sort,timezone,view}"
    }
  ]
}
]
}
```

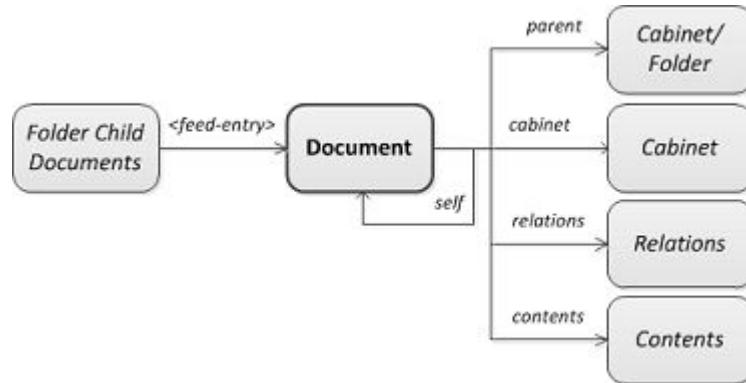


## 2.24 Document

The Document resource represents a document in a repository.

### 2.24.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder Child Documents](#)” on page 240 and [About the Diagram](#) on page 13.

### 2.24.2 Feed

No feed support.

### 2.24.3 Link relations

Link relation	Description	Resource reference
self	This document	<a href="#">“Document” on page 214</a>
parent	The parent folder (or cabinet) of this document	<a href="#">“Cabinet” on page 132</a> <a href="#">“Folder” on page 233</a>
cabinet [1]	The cabinet of this document	<a href="#">“Cabinet” on page 132</a>
relations [1]	A collection of relations related to this document	<a href="#">“Relations” on page 538</a>
contents	A collection of contents associated to this document	<a href="#">“Contents” on page 178</a>
comments	The comments associated to this document	N/A
related-documents	The collection of documents related to this document by way of a given relation	N/A

Link relation	Description	Resource reference
version-history	The collection of versions for this document	N/A
virtual-document-nodes	<p>Virtual document nodes displayed when the <code>SysObject</code> is a virtual document in the following cases:</p> <ul style="list-style-type: none"> <li>• In 7.3.0, it is displayed only when the document is a virtual document</li> <li>• Since 7.3.1, all documents have this link relation because users can convert a simple document into a virtual document by adding a new virtual document node</li> </ul>	<a href="#">“Virtual Document Nodes” on page 821</a>
snapshots	Link to the Virtual Document Snapshots resource.	<a href="#">“Virtual Document Snapshots” on page 839</a>
related-docs [1]	Allows you to find related or similar documents according to common knowledge among documents, such as keyword, hashtag, or category, etc.	<a href="#">“Related Documents” on page 542</a>
related-sme [1]	Allows you to find related SMEs (Authors) who have worked on any given document.	<a href="#">“Related SME” on page 550</a>
significant-terms [1]	Allows you to use key words while searching for content with Documentum Contextual Content Engine.	<a href="#">“Significant Terms” on page 670</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

## 2.24.4 Operations

### 2.24.4.1 Supported HTTP methods

Method	Description
GET	Retrieves properties, and other information of the document resource.
POST	Updates the properties for the document resource.
DELETE	Deletes the document resource from a repository.

### 2.24.4.2 Get a document

GET properties and other information of the document resource. Properties are returned as embedded elements in the response message body. Other information, such as relationships, is referenced from the link relations of the response message body.

#### 2.24.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	check-subscription	Determines whether or not to check if the current user is subscribed to the requested instance.	string	<i>true, false</i>	<i>false</i>

#### 2.24.4.2.2 Request media types

N/A

#### 2.24.4.2.3 Request headers

- Accept
- Authorization

**2.24.4.2.4 Request body**

N/A

**2.24.4.2.5 Response headers**

- Content-Type

**2.24.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

**2.24.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.24.4.2.8 Response body**

 **Example 2-79: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://identifiers.emc.com/vocab/documentum"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:type="dm_document" definition="http://localhost:8080/dctm-rest/
           repositories/
             REPO/types/dm_document>
   <doc-type-label>SIMPLE_DOCUMENT</doc-type-label>
   <properties>
     <attr1>value1</attr1>
     <repeating_attr2>
       <item href="xxx2">value2</item>
       <item href="xxx3">value3</item>
     </repeating_attr2>
     ...
   </properties>
   <link rel="self" href="..." />
   ...
</dm_document>
```



 **Example 2-80: JSON response**

```
{
  "name": "document",
  "type": "dm_document",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
```

```
dm_document",
"doc-type-label": "SIMPLE_DOCUMENT",
"properties":
{
    "attr1": "value1",
    "repeating_attr2": [value2, value3],
    ...
},
"links":
[
    {
        "rel": "self",
        "href": "..."
    },
    ...
}
```



#### 2.24.4.3 Update a document

Update the document with given properties. When the client provides undefined properties in the Request body, the server throws an exception.

##### 2.24.4.3.1 HTTP Request method and query parameters

PUT

##### 2.24.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.24.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.24.4.3.4 Request body

XML or JSON representation of the document resource. Only updatable properties can be put in the message body.

##### ➡ Example 2-81: XML request body

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://identifiers.emc.com/vocab/documentum"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <doc-type-label>VIRTUAL_DOCUMENT</doc-type-label>
    <properties>
        <attr1>value1</attr1>
        <repeating_attr2>
            <item href="xxx2">value2</item>
            <item href="xxx3">value3</item>
        </repeating_attr2>
        ...
    </properties>
</dm_document>
```



##### ➡ Example 2-82: JSON request body

```
{
    "name": "document",
    "type": "dm_document",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document",
    "doc-type-label": "VIRTUAL_DOCUMENT",
    "properties":
    {
        "attr1": "value1",
        "repeating_attr2": [value2, value3],
        ...
    }
}
```



#### 2.24.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.24.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.24.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.24.4.3.8 Response body

A successful response contains all of the properties of the updated document.

##### Example 2-83: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://identifiers.emc.com/vocab/documentum"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:type="dm_document" definition="http://localhost:8080/dctm-rest/
repositories/
REPO/types/dm_document">
<doc-type-label>VIRTUAL_DOCUMENT</doc-type-label>
<properties>
    <attr1>value1</attr1>
    <repeating_attr2>
        <item href="xxx2">value2</item>
        <item href="xxx3">value3</item>
    </repeating_attr2>
    ...
    <properties>
        <link rel="self" href="..." />
    ...
</dm_document>
```



##### Example 2-84: JSON response

```
{
    "name": "document",
    "type": "dm_document",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document",
    "doc-type-label": "VIRTUAL_DOCUMENT",
    "properties":
    {
        "attr1": "value1",
        "repeating_attr2": [value2, value3],
        ...
    },
}
```

```
    "links":  
    [  
        {  
            "rel": "self",  
            "href": "..."  
        },  
        ...  
    ]  
}
```



#### 2.24.4.4 Delete a document

Deletes the Document resource from the repository.

When the deletion fails, an exception is thrown and the Document resource is reverted to the original state.

##### 2.24.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	del-version	<p>Delete options for multi-version document objects.</p> <ul style="list-style-type: none"> <li>• SELECTED VERSION: The version associated with the given object ID is removed.</li> <li>• unused : The versions that do not have symbolic labels associated with them are removed.</li> <li>• ALL_VERSIONS : All revisions on the same version tree are removed.</li> </ul>	string	N/A	all

HTTP method	Query parameters	Description	Data type	Value range	Default value
	del-vd-all	<p>Delete the virtual document and all of its descendants</p> <ul style="list-style-type: none"> <li>• <i>false</i>: Deletes only the virtual document and the containments with its direct children</li> <li>• <i>true</i>: Deletes the virtual document and all of the descendants of this virtual document</li> </ul>	string	<i>true, false</i>	<i>false</i>

#### 2.24.4.4.2 Request media types

N/A

#### 2.24.4.4.3 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.24.4.4.4 Request body

N/A

#### 2.24.4.4.5 Response headers

N/A

#### 2.24.4.4.6 Response media types

N/A

#### 2.24.4.4.7 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.24.4.8 Response body

HTTP 204 No Content Success status upon a successful delete operation. The response body contains no content.

### 2.24.4.5 Example

➡ **Example 2-85: Delete a document**

*Request*

```
DELETE http://localhost:8080/dctm-rest/repositories/REPO/documents/
0900000f80000cd8
Accept:application/vnd.emc.documentum+json
Authorization:Basic ZG1hZG1pbjpwYXNzd29yZA==
```

*Response*

```
Status Code: 204 No Content Success
```

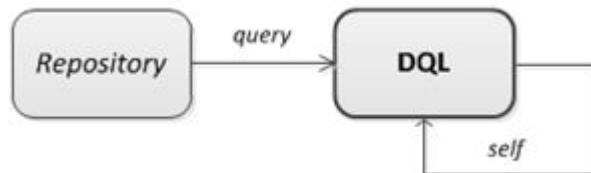


## 2.25 DQL

The DQL resource represents a read-only DQL query.

## 2.25.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

## 2.25.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: DQL Query Results
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: Query result corresponding to one of the rows in the result set
- Entry ID: Query resource URI with the item index in the current page
- Entry Title: Value of the first property of the query result. For example, if you perform the DQL statement `select r_object_id, object_name from dm_document`, the value of `r_object_id` is used as the title.
- Entry Updated: `r_modify_date` of the query result if this property exists. Otherwise, an empty string.

## 2.25.3 Link relations

Link relation	Description	Resource reference
self	This DQL resource	<a href="#">“DQL” on page 224</a>
first, last, next, previous	Pagination links.	<a href="#">“DQL” on page 224</a>

## 2.25.4 Operations

### 2.25.4.1 Supported HTTP methods

Method	Description
GET	Performs a read-only DQL query with the <code>dql</code> query parameter.
POST	Performs a read-only DQL query with the <code>dql</code> query parameter or with the Request body encoded in <code>application/x-www-form-urlencoded</code> .

### 2.25.4.2 Perform a read-only DQL query

Perform a read-only DQL query with the `dql` query parameter.

Using the GET operation to perform a query can result in a response with better cache performance. However, the web browser or client library may fail to handle the Request properly when the DQL query is very long (for example, over 2,000 characters) since the DQL statement is provided using the HTTP query parameter `dql`. To perform a query with a long DQL statement, use the POST operation.

For more information, see [Perform a long read-only DQL query](#).

#### 2.25.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Default value	Yes
GET	<code>dql</code>	The query parameter of the DQL resource	string	null This parameter is required.	Yes

HTTP method	Query parameters	Description	Data type	Default value	Yes
	raw	<p>Indicates whether to return the DQL results as raw property bags or with links generated.</p> <p>When this parameter is set to <i>false</i>, the REST server generates link relations and/or including thumbnail links if possible.</p>	string	<i>false</i>	No

 **Note:** International characters that are used in query parameters must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.

Also, this operation supports the following common query parameter:

- `page`
- `items-per-page`

 **Note:** This operation does not support the `include-total` query parameter. To obtain the total size of the result set, send another query by using the aggregation function `COUNT()` with the same conditions.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.25.4.2.2 Request media types

N/A

#### 2.25.4.2.3 Request headers

- Accept
- Authorization

For more information about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.25.4.2.4 Request body

N/A

#### 2.25.4.2.5 Response headers

- Content-Type

For more information about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.25.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.25.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.25.4.2.8 Response body

XML or JSON representation of result set of the DQL query.

##### Example 2-86: XML response

```
...
<entry>
  <id>http://localhost:8080/dctm-rest/repositories/acme01?dql=...</id>
  <title>090004d2800001d0</title>
  <updated>2013-06-18T20:17:39.331+08:00</updated>
  <content>
    <dm:query-result>
      <dm:properties>
        <dm:r_object_id>090004d2800001d0</dm:r_object_id>
        <dm:object_name>Default Signature Page Template</dm:object_name>
      </dm:properties>
    </dm:query-result>
```

```
</content>
</entry>
...

```



#### ➤ Example 2-87: JSON response

```
...
{
    "id": "http://localhost:8080/dctm-rest/repositories/acme01?dql=...",
    "title": "090004d2800001d0",
    "updated": "2013-06-18T20:22:41.425+08:00",
    "content": {
        "name": "query-result",
        "properties": {
            "r_object_id": "090004d2800001d0",
            "object_name": "Default Signature Page Template"
        }
    },
    ...
}
```



- The query result is included in the `query-result` element.
- If the client tries to perform a non-select query, the server returns an HTTP 400 Bad Request status.
- The returned query result set only contains the resources that you have access to.
- Pagination is supported.



**Note:** The query result may contain link relations or thumbnail links. For more information, see *OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)*.

#### 2.25.4.3 Perform a long read-only DQL query

Perform a read-only DQL query with the `dql` query parameter or with the request body encoded in `application/x-www-form-urlencoded`.

To perform a query with long DQL statement, use the POST operation with a request body in `application/x-www-form-urlencoded`.

#### 2.25.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	dql	Read-only DQL statement to perform	string	N/A	null This parameter is required if the HTTP request does not contain a body.
	raw	Indicates whether or not to return the DQL results as raw property bags or with links generated.  When this parameter is set to <i>false</i> , the REST server generates link relations and/or including thumbnail links if possible.	string	<i>true, false</i>	<i>false</i>

- `page`
- `items-per-page`

This operation also supports the following common query parameter:



**Note:** This operation does not support the `include-total` query parameter. To obtain the total size of the result set, send another query using the aggregation `COUNT()` function with the same conditions.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.25.4.3.2 Request media types

- application/x-www-form-urlencoded

#### 2.25.4.3.3 Request headers

- Accept
- Authorization

For more information about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.25.4.3.4 Request body

Query statement to perform in application/x-www-form-urlencoded. Alternatively, you can leave the request body blank and use the `dql` parameter to hold the query statement. Use the request body when the query statement is long.



**Note:** An error may occur if you specify the same query statement in both the request body and the `dql` parameter.

#### ➤ Example 2-88: Example request

```
Accept: application/vnd.emc.documentum+json
Content-Type: application/www-form-urlencoded
POST http://localhost:8080/dctm-rest/repositories/acme01
dql=select%20r_object_id,object_name%20from%20dm_document&items-per-
page=100&page=1
```



#### 2.25.4.3.5 Response headers

- Content-Type

For more information about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.25.4.3.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.25.4.3.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.25.4.3.8 Response body

XML or JSON representation of result set of the DQL query.

##### Example 2-89: XML response

```
...
<entry>
  <id>http://localhost:8080/dctm-rest/repositories/acme01?dql=...</id>
  <title>090004d2800001d0</title>
  <updated>2013-06-18T20:17:39.331+08:00</updated>
  <content>
    <dm:query-result>
      <dm:properties>
        <dm:r_object_id>090004d2800001d0</dm:r_object_id>
        <dm:object_name>Default Signature Page Template</dm:object_name>
      </dm:properties>
    </dm:query-result>
  </content>
</entry>
...
```



##### Example 2-90: JSON response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/acme01?dql=...",
  "title": "090004d2800001d0",
  "updated": "2013-06-18T20:22:41.425+08:00",
  "content": {
    "name": "query-result",
    "properties": {
      "r_object_id": "090004d2800001d0",
      "object_name": "Default Signature Page Template"
    }
  }
},
```



- The query result is included in the `query-result` element.
- If the client tries to perform a non-select query, the server returns an HTTP 400 Bad Request status.

- The returned query result set only contains the resources that you have access to.
- Pagination is supported.

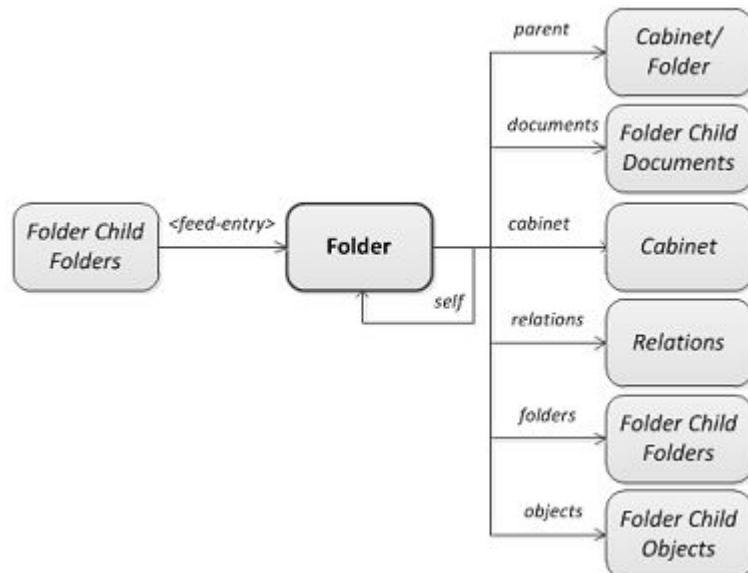
 **Note:** The query result may contain link relations or thumbnail links. For more information, see *OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)*.

## 2.26 Folder

The `Folder` resource represents a folder in a repository.

### 2.26.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



See Also: “[Folder Child Folders](#)” on page 268 and [About the Diagram](#) on page 13.

### 2.26.2 Feed

No feed support.

## 2.26.3 Link relations

Link relation	Description	Resource reference
self	Link to this folder.	<a href="#">"Folder" on page 233</a>
parent	Link to the parent folder (or cabinet) of this folder.	<a href="#">"Cabinet" on page 132</a> <a href="#">"Folder" on page 233</a>
relations [1]	Collection of relations related to this folder.	<a href="#">"Relations" on page 538</a>
cabinet [1]	Link to the cabinet of this folder.	<a href="#">"Cabinet" on page 132</a>
folders [1]	Collection of folders linked to this folder.	<a href="#">"Folder Child Folders" on page 268</a>
documents [1]	Collection of documents linked to this folder.	<a href="#">"Folder Child Documents" on page 240</a>
objects [1]	Collection of SysObjects linked to this folder.	<a href="#">"Folder Child Objects" on page 282</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.26.4 Operations

### 2.26.4.1 Supported HTTP methods

Method	Description
GET	Retrieves properties, and other information of the Folder resource.
POST	Updates the properties for the Folder resource.
DELETE	Deletes the Folder resource from a repository.

### 2.26.4.2 Get a Folder

GET properties, and other information, such as relationships, of the Folder resource. Properties are returned as embedded elements in the response message body. Other information, such as relationships, is referenced from the link relations of the response message body.

#### 2.26.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	links				
	check-subscription	Determines whether or not to check if the current user is subscribed to the requested instance.	string	<i>true, false</i>	<i>false</i>

#### 2.26.4.2.2 Request media types

N/A

#### 2.26.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.26.4.2.4 Request body

N/A

#### 2.26.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.26.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.26.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.26.4.2.8 Response body**

XML or JSON message that represents the folder.

### **2.26.4.3 Update a folder**

Update the folder with given properties. If the client provides undefined properties in the Request body, the server throws an exception.

#### **2.26.4.3.1 HTTP Request method and query parameters**

POST

#### **2.26.4.3.2 Request media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### **2.26.4.3.3 Request headers**

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.26.4.3.4 Request body**

XML or JSON representation of the Folder resource. Only updatable properties can be put in the message body.

#### 2.26.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.26.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.26.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.26.4.3.8 Response body

XML or JSON representation of the folder. The successful response contains all properties of the updated folder.

### 2.26.4.4 Delete a Folder

Deletes the `Folder` resource from the repository.

Folder children deletion is controlled by query parameters.

If the deletion fails, an exception is thrown and the `Folder` resource (including the folder tree and version history) is reverted to the original state.

#### 2.26.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	del-non-empty	<p>Specifies whether this operation deletes a non-empty folder or not.</p> <ul style="list-style-type: none"><li>• <i>true</i> - The folder and all its descendants are deleted.</li><li>• <i>false</i> - Only this folder is deleted. Deleting a non-empty folder results in an error.</li></ul>	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	del-all-links	<p>Specifies whether a multi-linked descendant is deleted or unlinked from this specified folder.</p> <ul style="list-style-type: none"> <li>• <i>true</i> - This operation deletes a multi-linked descendant along with all its folder links.</li> <li>• <i>false</i> - This operation only deletes a descendant's link to this folder. The descendant is still linked to other folders.</li> </ul> <p> <b>Note:</b> Virtual documents are not supported.</p>	string	<i>true, false</i>	<i>false</i>

#### **2.26.4.4.2 Request media types**

N/A

#### **2.26.4.4.3 Request headers**

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.26.4.4.4 Request body**

N/A

#### **2.26.4.4.5 Response headers**

N/A

#### **2.26.4.4.6 Response media types**

N/A

#### **2.26.4.4.7 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.26.4.4.8 Response body**

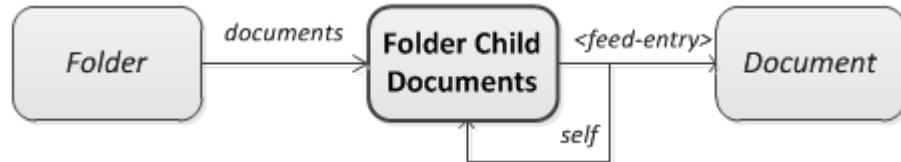
HTTP 204 No Content status upon a successful delete operation. The Response body contains no content.

## **2.27 Folder Child Documents**

The Folder Child Documents resource represents the collection of all documents, which can be any object of the dm\_document type or its subtype, under a specified folder.

## 2.27.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder](#)” on page 233 and [About the Diagram](#) on page 13.

## 2.27.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the documents without the file extension
- Feed Title: List of folder child documents
- Feed Updated: Server's current time
- Entry: A [Document](#) object filed under this folder
- Supports POST? Yes

## 2.27.3 Link relations

Link relation	Description	Resource reference
self	The URI for folder child documents collection feed	<a href="#">“Folder Child Documents” on page 240</a>
first, last, next, previous	Pagination links	<a href="#">“Folder Child Documents” on page 240</a>

## 2.27.4 Operations

### 2.27.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a list of all documents under the specified folder
POST	Creates a new document under the specified folder

### 2.27.4.2 Get all documents of a folder

This operation retrieves a collection of all the child documents of a specified folder and according to the folder object type. You can specify any folder type to retrieve from the parent folder, including a `dm_folder` type, a `dm_cabinet` type, or a custom type. If the parent folder contains documents of the specified type and you have permissions to access those documents, they will be retrieved.



**Note:** To maintain security, the REST server will only retrieve documents that the logged in user has permissions to access.

#### 2.27.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	hide-shared-parent	Specifies whether to show the shared parent object of the lightweight object.  <b>Note:</b> Enabling <code>hide-shared-parent</code> may decrease performance.	string	<code>true, false</code>	<code>false</code>
	object-type	Specifies the name of a subtype of <code>dm_document</code> and returns only documents of that subtype.	string	Any sub type name of <code>dm_document</code>	null

This method supports the following URI request query parameters:

- `inline`
- `sort`



**Note:** The default sort order is by creation date, descending from most recent to oldest.

- page
- items-per-page
- include-total
- view
- links
- q: Provides support for full text searching using a subset of the Simple Search Language, however the parenthesis is not supported.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.27.4.2.2 Request media types

N/A

#### 2.27.4.2.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.27.4.2.4 Request body

N/A

#### 2.27.4.2.5 Response headers

- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.27.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.27.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.27.4.2.8 Response body

XML or JSON representation of the collection of child objects under the specified folder.

- The body must contain a list of document resources, each of which is filed under the specified folder.
- Each object may contain all properties of the object, depending on the setting of the `view` query parameter.
- Each object may contain links that are specific to the **Document** type that is filed in the specified folder.

See “[Link relations](#)” on page 693 in the `SysObject` resource, which provides information on what links are available for various object types.

- The retrieved child documents collection only contains those objects that the user has permissions to access.

The following code samples show you typical requests along with responses.

##### ➡ Example 2-91: XML request

```
GET http://localhost/rest/repositories/repo/folders/0c00000c80000105/documents
Accept: application/vnd.emc.documentum+xml
```



##### ➡ Example 2-92: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/repo/folders/
  0c00000580000107/
  documents</id>
  <title>Documents</title>
  <author>
```

```

<name>OpenText Documentum</name>
</author>
<updated>2018-01-11T02:32:49.139+00:00</updated>
<dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
<dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    1
</dm:items-per-page>
<link rel="self"
      href="http://localhost:8080/dctm-rest/repositories/repo/folders/
          0c00000580000107/documents?inline=true&amp;items-per-page=1"/>
<link rel="next"
      href="http://localhost:8080/dctm-rest/repositories/repo/folders/
          0c00000580000107/documents?inline=true&amp;items-per-
page=1&amp;page=2"/>
<link rel="first"
      href="http://localhost:8080/dctm-rest/repositories/repo/folders/
          0c00000580000107/documents?inline=true&amp;items-per-
page=1&amp;page=1"/>
<entry>
    <id>
        http://localhost:8080/dctm-rest/repositories/repo/documents/
0900000580006435
    </id>
    <title>obj_without_content</title>
    <author>
        <name>dmadmin</name>
        <uri>http://localhost:8080/dctm-rest/repositories/repo/users/dmadmin</
uri>
    </author>
    <summary>dm_document 0900000580006435</summary>
    <updated>2018-01-08T09:23:27.000+00:00</updated>
    <published>2018-01-08T09:23:27.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/repo/documents/
              0900000580006435"/>
    <content>
        <dm:document
            definition="http://localhost:8080/dctm-rest/repositories/repo/
                types/dm_document"
            xmlns:dm="http://identifiers.emc.com/vocab/documentum"
            xsi:type="dm_document">
            <dm:properties>
                <dm:r_object_id>0900000580006435</dm:r_object_id>
                <dm:object_name>my doc</dm:object_name>
                <dm:owner_name>dmadmin</dm:owner_name>
                ...
            </dm:properties>
            <dm:links>
                <dm:link rel="self"
                      href="http://localhost:8080/dctm-rest/repositories/repo/
documents/
                      0900000580006435"/>
                ...
            </dm:links>
        </dm:document>
    </content>
</entry>
</feed>
```



► **Example 2-93: JSON request**

```
GET http://localhost/rest/repositories/repo/folders/0c00000c80000105/documents  
Accept: application/vnd.emc.documentum+json
```



► **Example 2-94: JSON response**

```
{  
    "id": "http://localhost:8080/rest/repositories/repo/folders/0c00000c80000105/  
        documents",  
    "title": "Documents",  
    "updated": "2018-01-11T02:32:49.139+00:00",  
    "offset": 0,  
    "limit": 50,  
    "links":  
    [  
        {  
            "rel": "self",  
            "href": "http://localhost:8080/rest/repositories/repo/folders/  
                0c00000c80000105/documents"  
        },  
        {  
            "rel": "first",  
            "href": "http://localhost:8080/rest/repositories/repo/folders/  
                0c00000c80000105/documents?offset=0&limit=50"  
        }  
    ],  
    "entries":  
    [  
        {  
            "id": "http://localhost:8080/rest/repositories/repo/documents/  
                6700000c8000010a",  
            "title": "obj_without_content",  
            "updated": "2018-01-08T09:23:27.000+00:00",  
            "content":  
            {  
                "properties":  
                {  
                    "r_object_id": "6700000c8000010a",  
                    "object_name": "my doc",  
                    "owner_name": "dmadmin",  
                    ...  
                },  
                "links":  
                [  
                    {  
                        "rel": "self",  
                        "href": "http://localhost:8080/rest/repositories/repo/documents/  
                            6700000c8000010a"  
                    }  
                    ...  
                ]  
            },  
            ...  
        }  
    ]  
}
```



### 2.27.4.3 Create a document in a folder

This operation creates a child document in the specified folder. You can set the properties of the object while creating the object. The object must be a `dm_document` object type or one of its subtypes. By default, the object type is `dm_document`.

#### 2.27.4.3.1 HTTP Request method and query parameters

When storing content to Centera storage, the `content-length` query parameter must be provided. When specifying the value for `content-length`, you must use an accurate byte count of the content to be uploaded.

For other storage mediums, such as hard disk, the `content-length` parameter is not required and is optional. However, even when not required, if this parameter is specified, it must be accurate.



**Note:** When creating a document that contains no content, neither the `content-length` or the `content charset` query parameters are used. These two query parameters are only used when the documents that are being created or filed contain content.

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	<code>r_object_type</code>	Specifies the object type of the new object being created.	string	For this resource, any valid <code>dm_document</code> type or one of its subtypes	<code>dm_document</code>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-length	<p>Specifies the byte count of the content to be uploaded</p> <p>When storing content to Centera storage, the content-length parameter must be provided. When providing the value for content-length, you must provide an accurate byte count of the content to be uploaded.</p> <p>For other storage mediums, such as harddisk, the content-length parameter is optional. However, even when not required, if this parameter is provided, it must be accurate.</p>	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-charset	<p>Used to sanitize content.</p> <p>This parameter tells the server how to parse the content for sanitizing. When the content-charset is not provided, the server tries to get metadata information from the content itself, as long as the content is in HTML format.</p> <p>When the metadata has a valid charset, then it is used to sanitize the content. The configuration default charset is used when the content-charset parameter and the metadata information are not found</p> <p> <b>Note:</b> When the charset and the content do not</p>	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
		match, the uploaded content may have an incorrectly encoded value.			

#### 2.27.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.27.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.27.4.3.4 Request body

XML or JSON representation of the document that will be created.

- Object type of the new document is specified in the `r_object_type` property of the Request body. The `r_object_type` property is optional. If this property is not specified in a client request, it is set to `dm_document` by default.
- Documentum CM Server validates the constraints and permissions on setting the other properties, including any read-only properties.
- Any whitespace in the value for the `r_object_type` property is trimmed away by the server.

#### 2.27.4.3.4.1 Create a document with aspects and a lifecycle

This section shows a code sample that demonstrates how to create a document, attach Aspects to it, and then attach a Lifecycle. During this operation, the document object is created first, then once the document object exists, the Aspects are attached to it, and finally the Lifecycle is attached to it. If the operation to create this document with aspects and a lifecycle fails, the newly created document object must be destroyed. In this case, the REST server will attempt to drop the document object, however, if this attempt fails, then the document object must be dropped manually, otherwise it remains as an orphaned object in your system.

When you create a document object in this manner you can also set its properties in addition to the `object-aspects` and `object-lifecycle`. You can set the properties by using the `aspect_name.property_name` parameter. When specifying the aspect in the `properties` attribute, the aspects to be attached in the `object-aspects` attribute will be ignored.



#### Caution

You cannot set Aspect properties using the `properties` attribute when you specify that a document will be created with a `Lifecycle` as part of the procedure. This limitation occurs because this procedure does not support the use of a transaction when adding a `Lifecycle` to the object and also because when a `Lifecycle` is being attached, the REST server creates the document with its Aspects first, then it attaches the `Lifecycle` to the document. This means that the document object will have its Aspects attached to it before the aspect properties can be set using the `properties` attribute. When you intend to attach a `Lifecycle` to your document object, use the `aspect_name.property_name` parameter to set your Aspect properties.

#### Example 2-95: XML request

```
<document>
  <properties>
    <object_name>test_doc</object_name>
    <r_object_type>dm_document</r_object_type>
    <xxx_aspect.xyy>...</xxx_aspect.xyy>
    <xxx_aspect.yyy>
      <item>123456</item>
    </xxx_aspect.yyy>
  </properties>

  <object-lifecycle>
    <lifecycle-id>xxx</lifecycle-id>
    <current-state>xxx</current-state>
    <aliasasset-name>xxx</aliasasset-name>
  </object-lifecycle>

  <object-aspects>
    <aspect>xxx</aspect>
  </object-aspects>
</document>
```



 **Example 2-96: JSON request**

```
{  
    "properties" : {  
        "object_name" : "test doc",  
        "r_object_type" : "dm_document",  
        "xxx_aspect.xxx" : "...",  
        "xxx_aspect.yyy" : [  
            123456  
        ]  
    },  
  
    "object-lifecycle" : {  
        "lifecycle-id" : "xxx",  
        "current-state" : "xxx",  
        "aliasset-name" : "xxx"  
    },  
  
    "object-aspects" : {  
        "aspects": [  
            "xxx_aspect"  
        ]  
    }  
}
```

**2.27.4.3.5 Response headers**

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.27.4.3.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

**2.27.4.3.7 Response status**

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.27.4.3.8 Response body

XML or JSON representation of the document created under the specified folder.

- Each object contains all properties of a document.
- Each object contains links specific to the document object type.

For more information on what kind of links are available for a certain type of object, see “[Link relations](#)” on page 693 in the `SysObject` resource.

#### ➤ Example 2-97: XML request

```
POST http://localhost/rest-api-web/repositories/repo/folders/0c00000c80000105/documents
Content-Type: application/vnd.emc.documentum+xml

<document>
  <properties>
    <object_name>obj_without_content</object_name>
    <r_object_type>dm_document</r_object_type>
    ...
  </properties>
</document>
```



#### ➤ Example 2-98: XML response

```
Status Code: 201 Created
Location: http://localhost/rest/repositories/repo/documents/0900000d10000983
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<document>
  <properties>
    <object_name>testDocument</object_name>
    <r_object_type>dm_document</r_object_type>
    <keywords>"rest", "documentum"</keywords>
    ...
  </properties>
  <links>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/documents/0900000580006435"/>
    ...
  </links>
</document>
```



#### ➤ Example 2-99: JSON request

```
POST http://localhost/rest-api-web/repositories/repo/folders/0c00000c80000105/documents
Content-Type: application/vnd.emc.documentum+json

{
  "attributes": {
    "object_name": "testDocument",
    "r_object_type": "dm_document",
```

```
        "keywords": [ "rest", "documentum" ]
        ...
    }
```



#### ➤ Example 2-100: JSON response

```
Status Code: 201 Created
Location: http://localhost/rest/repositories/repo/documents/0900000d10000983
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "attributes":
  {
    "r_object_id": "0900000d10000983",
    "object_name": "testDocument",
    "r_creation_date": "2012-06-13T05:36:08.000+0800",
    "r_modify_date": "ck"2012-06-13T05:36:08.000+0800",
    .....
  },
  "links":
  [
    {
      "rel": "self",
      "href": "http://localhost:8080/rest/repositories/repo/documents/
              0900000d10000983"
    }
  ]
}
```



### 2.27.4.4 Import document metadata and content

This operation creates a document and its contents in a specified folder in a single request.

#### 2.27.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	format	<p>Specifies the <code>dm_format</code> name for the contents.</p> <p>When more than one type of content is being imported, the <code>format</code> parameter can be used to define the format for each piece of content by using a comma separated string.</p> <p>For example, <code>format=crtext,html,pub_html</code></p> <p>For more information, see <a href="#">format</a>.</p>	string	Comma separated string	When this parameter is not specified, the REST server will try to determine the format using the Content-Type specified in the primary content part of the multipart Request. The REST server will assign this format to the primary content. However, this format is not guaranteed to be accurate.
	all-primary	Specifies whether to import all contents using the same type as the primary content for different pages (page 0, 1, 2...), or one primary content, and others as renditions for the first page, which is page 0	string	true or <i>false</i>	<i>true</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	modifier	<p>Specifies the modifier of the contents. For the primary content, the modifier is ignored.</p> <p>For example, when you import with the all-primary parameter set to <i>false</i>, the value of the modifier can be:  <i>modifier=</i>, mod1,mod2, mod3. That means the first (or primary) content has no modifier, and the second rendition's modifier is mod1, the third is mod2 and so on.</p> <p>Setting the all-primary parameter to <i>true</i>, or when <i>content-count&lt;=1</i> causes all modifiers to be ignored.</p>	string	Comma separated string	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-length	<p>Specifies the byte count of the content to be uploaded. This parameter is optional.</p> <p>A comma separated list can be used to define multiple values for this parameter. For example: content-length=1024,2048,,4096. The blank value shown here is 0.</p> <p>For more information, see <a href="#">content-length</a>.</p>	string	Comma separated string	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-charset	This parameter is used to sanitize content and it defines how the server parses the content while sanitizing. Multiple values for this parameter can be set using a comma separated list. For example: content-charset=UTF-8,GBK,ISO-8859-1.  For more information, see <a href="#">content-charset</a> .	string	Comma separated string	null



**Note:** Here are some additional details about the query parameters listed above:

- **format**
  - The `format` list must match the content sequence. When you don't want to specify the format for a piece of content you can leave it blank in the comma separated list but you must still include the comma as a placeholder. For example: `format=crtext,,html`
  - When the format for contents is not defined, the server uses the content media type to select the format.
  - When the `all-primary` parameter is `true`, all primary content types must have the same format. In this case, when you define more than one format, they must be the same. For example: `format=crtext,crtext,crtext`. You can also define one format for all of the primary content: `format=crtext`
- **content-charset**
  - When the metadata has a valid charset, then it is used to sanitize the content.

- When the `content-charset` is not provided and the content is in HTML format, the server tries to get a value for the `content-charset` parameter from the metadata information of the content itself.
  - The configuration default charset is used when the `content-charset` parameter and the metadata information are not found.
  - When the charset and the content do not match, the uploaded content may have an incorrectly encoded value.
- `content-length`
    - When the `all-primary` parameter is set to `false`, the first value of in the `content-length` parameter is used for the primary piece of content, the other values in the comma separated string are ignored.
    - When storing content to Centera storage, the `content-length` parameter is mandatory and must be provided. For other storage mediums, such as harddisk, the `content-length` parameter is optional.
    - In all cases, including when the `content-length` parameter is not required, if a value for it is set, the value must be an accurate byte count of the contents to be uploaded.
  - `content-count`
    - When you import more than one piece of contents, the REST server must know how many content pieces there are in total. The total number of content pieces cannot be obtained until the end of the multipart stream. Therefore, the REST client must specify this number in the value of the `content-count` parameter.
    - When the value set in the `content-count` parameter is greater than the actual number of content pieces, an exception is thrown because the system expects more content pieces than it has received.
    - When the value set in the `content-count` parameter is less than the actual number of content pieces, all content pieces that exist in addition to the number of pieces declared in the `content-count` parameter are ignored.



### Caution

Ensure that the value provided by the `content-count` parameter is accurate.

#### 2.27.4.4.2 Request media types

- multipart/form-data
- multipart/mixed

#### 2.27.4.4.3 Request headers

- Accept
- Authorization
- Content-Type



##### Caution

The Content-Type header must be either multipart/form-data or multipart/mixed, which notifies the REST server that this request is a multipart request.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.27.4.4.4 Request body

The request body is a multipart message that must conform to RFC 2046. A multipart/form-data or multipart/mixed message that contains a series of parts. This series must conform to the following rules:

- The series must contain two parts, one metadata part and one contents parts.
- The first part in the series must contain the metadata for the document to be created.

Content-Type for the first part must be one of the following values:

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

If the client does not set the Content-Type value or if the client sets the Content-Type to an invalid value, then the REST server uses application/vnd.emc.documentum+json to overwrite the value.

- The second part, which is the content part, in the series must be the binary data of the primary contents. Its Content-Type must be the MIME type for the primary content.
- Each part in the series must contain a Content-Disposition header. The disposition type of the header must be form-data. The disposition must contain a parameter called name that specifies the name of this part. The name parameter may be ignored by REST server. This behavior conforms to the W3 rules for HTML multipart forms and is a decision that should be made from both the server implementation and the client consumption perspective.



**Note:** Most libraries for various programming languages can handle the multipart processing of an HTML form. However, the generic multipart processing library is not prevalent. Therefore, we selected the HTML multipart/form-data Content-Type to include a broader audience of compatibility.

- This operation also supports the usage of Aspects and Lifecycles. For more information, see [Create a Document With Aspects and a Lifecycle](#).

➡ **Example 2-101: XML request - multipart/form-data or multipart/mixed**

```
--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+xml

<object>
  <properties>
    <object_name>xyz</object_name>
  </properties>
</object>

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the first content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the second content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the third content part

--314159265358979--
```



➡ **Example 2-102: JSON request - multipart/form-data or multipart/mixed**

```
--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+json
{
  "properties": {
    "object_name": "xyz"
  }
}

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the first content part
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the second content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the third content part

--314159265358979--
```



### Example 2-103: Use multipart in a custom controller

```
@RequestMapping(method = RequestMethod.POST, consumes = { "multipart/form-data",
    "multipart/mixed" })
@ResponseStatus(HttpStatus.OK)
public void processMultipart(@RequestBody final Iterator<Part> parts) throws
DfException,
MissingServletRequestParameterException, IOException {
    if(parts != null) {
        response.setContentType(MediaType.TEXT_PLAIN.toString());
        OutputStream out = response.getOutputStream();
        int p = 1;

        while(parts.hasNext()) {
            Part part = parts.next();
            for(String header : part.getHeaderNames()) {
                // Write the part header to the Response directly
                out.write((header + "=" + part.getHeader(header)) + "\r
\n").getBytes();
            }
            // Write the part content to the Response
            IOUtils.copy(part.getInputStream(), out);
            out.write("\r\n-----end-----\r\n".getBytes());
            out.flush();
        }
    }
}
```

The sample reads the multipart input and writes it back to the response. Each part of the multipart input has headers, and the content stream. Each part can be individually read using one of the standard `<hasNext>` and `<next>` `<Iterator>` methods.

All of the parts have to be read sequentially because they are all contained within one multipart stream and the REST server does not cache it.



**Note:** When more than one content part exists, the REST server must know how many content parts there are in total. However, the total number of

content parts cannot be determined until the end of the multipart stream. Therefore, the REST client must provide the total number of content parts to the REST server by setting a value for the `content-count` parameter. For more information about this parameter, see [content-count](#).

#### 2.27.4.4.5 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.27.4.4.6 Response media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

#### 2.27.4.4.7 Response status

- 201 - Imported successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.27.4.4.8 Response body

XML or JSON representation of the document created under the specified folder.

- The body may contain all of the attributes of an object, depending on the setting of the `attr_filter` query parameter.
- The Document (or Document subtype) object may contain links that are specific to the Document object type or Document subtype object type.
- The response body must contain a link with the `enclosure` relation, which is the URI used to identify the primary content. A `length` attribute can be added to the link to indicate the length, in bytes, of the primary content.

#### ➡ Example 2-104: XML request

```
POST http://localhost/rest-api-web/repositories/repo/folders/
 0c00000c80000105/documents
Content-Type: multipart/form-data; boundary=314159265358979

--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+xml
<document>
  <properties>
    <object_name>document</object_name>
```

```
<r_object_type>dm_document</r_object_type>
...
</properties>
</document>

--314159265358979
Content-Disposition: form-data; name=binary1
Content-Type: text/plain

This is primary content
--314159265358979--
```



#### Example 2-105: XML response

```
Status Code: 201 Created
Location: http://localhost/rest/repositories/repo/documents/0900000d10000983
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<document>
  <properties>
    <object_name>obj_without_content</object_name>
    <r_object_type>dm_document</r_object_type>
    ...
  </properties>
  <links>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/
documents/
          0900000580006435" />
    ...
  </links>
</document>
```



#### Example 2-106: JSON request

```
POST http://localhost/rest-api-web/repositories/repo/folders/0c00000c80000105/
  documents
Content-Type: multipart/form-data; boundary=314159265358979

--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+json

{
  "attributes": {
    "object_name": "rest-api-test"
  }
}

--314159265358979
Content-Disposition: form-data; name=binary1
Content-Type: text/plain

This is primary content
--314159265358979--
```



### ➤ Example 2-107: JSON response

```
Status Code: 201 Created
Location: http://localhost/rest/repositories/repo/documents/0900000d10000983
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "attributes": {
    {
      "r_object_id": "0900000d10000983",
      "object_name": "testDocument",
      "r_creation_date": "2012-06-13T05:36:08.000+0800",
      "r_modify_date": "2012-06-13T05:36:08.000+0800",
      ...
    },
    "links": [
      {
        {
          "rel": "self",
          "href": "http://localhost:8080/rest/repositories/repo/documents/
0900000d10000983"
        },
        {
          "rel": "enclosure",
          "type": "text/plain",
          "href": "http://localhost:8080/rest/repositories/repo/documents/
0900000c80016641/content"
        }
      ]
    }
  }
}
```



#### 2.27.4.5 Create a document for distributed upload

This operation creates a document and adds the Distributed Content upload link so that the document object's primary content can be distributed to an Accelerated Content Services or Branch Office Caching Services server in a separate request. Distributing content in this manner improves the overall performance of data transfers on servers that are located at various geographic locations.

##### 2.27.4.5.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	r_object_type	Specifies the object type of the new object being created.	string	For this resource, any valid dm_document type or one of its subtypes	dm_document
	content-length	<p>Specifies the byte count of the content to be uploaded</p> <p>When storing content to Centera storage, the content-length parameter must be provided. When providing the value for content-length, you must provide an accurate byte count of the content to be uploaded.</p> <p>For other storage mediums, such as harddisk, the content-length parameter is optional. However, even when not required, if this parameter is provided, it must be accurate.</p>	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-charset	<p>Used to sanitize content.</p> <p>This parameter tells the server how to parse the content for sanitizing. When the content-charset is not provided, the server tries to get metadata information from the content itself, as long as the content is in HTML format.</p> <p>When the metadata has a valid charset, then it is used to sanitize the content. The configuration default charset is used when the content-charset parameter and the metadata information are not found</p> <p> <b>Note:</b> When the charset and the content do not match, the uploaded content may have an incorrectly encoded value.</p>	string	N/A	null

#### 2.27.4.5.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.27.4.5.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.27.4.5.4 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.27.4.5.5 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.27.4.5.6 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

## 2.28 Folder Child Folders

The Folder Child Folders resource represents the collection of all folders (an object of the dm\_folder type or its subtype) under a specified folder.

### 2.28.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder](#)” on page 233 and [About the Diagram](#) on page 13.

## 2.28.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the folders.
- Feed Title: List of folder child folders
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: “[Folder](#)” on page 233
- Entry ID: URI of the folder
- Entry Title: Folder name
- Entry Summary: Folder description
- Entry Updated: `r_modify_date` of the folder

## 2.28.3 Link relations

Link relation	Description	Resource reference
self	The URI for folder child objects collection feed.	“ <a href="#">Folder Child Folders</a> ” on page 268
first, last, next, previous	Pagination links.	“ <a href="#">Folder Child Folders</a> ” on page 268

## 2.28.4 Operations

### 2.28.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a list all child folders under the specified folder.
POST	Creates a new child folder under the specified folder.

### 2.28.4.2 Get the child folders under a folder

GET the collection of child folders under the given folder. You can specify any folder types, including `dm_folder`, `dm_cabinet` or a custom type.

#### 2.28.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	hide-shared-parent	Specifies whether to show the shared parent object of the lightweight object   <b>Note:</b> Enabling <code>hide-shared-parent</code> may decrease performance	string	<code>true</code> , <code>false</code>	<code>false</code>
	object-type	Specifies the exact sub type name to get from the collection of this type	string	N/A	null
	thumbnail	Specifies whether this operation deletes a non-empty folder or not. <ul style="list-style-type: none"><li>• <code>true</code> - Return the thumbnail link for each entry in the collection resource.</li><li>• <code>false</code> - Do not return the thumbnail link for each entry in the collection resource.</li></ul>	string	<code>true</code> , <code>false</code>	<code>false</code>

This method supports the following common query parameters:

- `inline`
- `sort`  
The default sort order is by creation date, descending (most recent to oldest).
- `page`
- `items-per-page`
- `include-total`
- `view`
- `links`
- `q`
- `filter`

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### **2.28.4.2.2 Request media types**

N/A

#### **2.28.4.2.3 Request headers**

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.28.4.2.4 Request body**

N/A

#### **2.28.4.2.5 Response headers**

- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.28.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.28.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

#### 2.28.4.2.8 Response body

XML or JSON representation of the collection of child objects under the specified folder.

- The body contains a list of child folder objects, each of which is filed under the specified folder.
- Each object may contain all properties of the folder, depending on the setting of the query parameter `view`.
- Each object contains links specific to the folder object type. See “[Link relations](#)” on page 693 in `SysObject` resource, which provides information on what kind of links are available for a certain type of object.
- The returned child objects collection only contains those that you have access to.

### 2.28.4.3 Create a child folder in a folder

Create a child folder in the specified folder. You can set the properties when creating the object. The object can be the type of `dm_folder` or its subtype other than `dm_cabinet` or its subtype. This is because `dm_cabinet` must be a top-level folder which cannot be filed under any other folder. By default, the object type is `dm_folder`.

#### 2.28.4.3.1 HTTP Request method and query parameters

POST

#### 2.28.4.3.2 Request media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

#### 2.28.4.3.3 Request headers

- `Accept`
- `Authorization`
- `Content-Type`

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.28.4.3.4 Request body

XML or JSON representation of the folder to create.

- The `object_name` property is required.
- Object type of the new document is specified in the `r_object_type` property of the request body.

The `r_object_type` property is optional. If this property is not specified in a client request, it is set to `dm_folder` by default.

- Documentum CM Server validates the constraints and permissions on setting the other properties (including read-only properties).

#### 2.28.4.3.5 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.28.4.3.6 Response media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`
- `application/xml`
- `application/json`

#### 2.28.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.28.4.3.8 Response body

XML or JSON representation of the folder created under the specified folder. The response contains all properties of the folder.

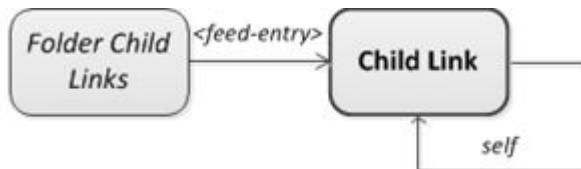
### 2.29 Folder Child Link (s)

#### 2.29.1 Folder Child Link

The Folder Child Link resource represents the folder containment relationship between a parent folder and one of its child objects.

##### 2.29.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder Child Links](#)” on page 277 and [About the Diagram](#) on page 13.

##### 2.29.1.2 Feed

No feed support.

##### 2.29.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this child link	<a href="#">“Folder Child Link (s)” on page 274</a>

##### 2.29.1.4 Operations

###### 2.29.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the folder link between the parent folder and its child object.
DELETE	Unlinks a given object from one of its parent folder.

#### 2.29.1.4.2 Retrieve a folder link

Retrieve the folder link between the child object and its parent folder.

##### 2.29.1.4.2.1 HTTP Request method and query parameters

GET

##### 2.29.1.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.29.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.29.1.4.2.4 Request body

N/A

##### 2.29.1.4.2.5 Response headers

- Content-Type

##### 2.29.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

##### 2.29.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.29.1.4.2.8 Response body

XML or JSON representation of the folder link between the object and its parent folder. The body contains the URI for the parent object and the URI for the child object. The body also contains the parent ID and child ID.

#### 2.29.1.4.3 Unlink an object from a parent folder

Unlink an object from one of its parent folders.



##### Notes

- To unlink an object from a primary link, you must have at least the write permission and the change location permission on the object.
- To unlink an object from a secondary link, you must have at least the browse permission and the change location permission on the object.
- If the repository is running under folder security, you must also have at least the write permission on the folder or cabinet from which the object is being unlinked.
- Documents and folders must have at least one link to a folder or cabinet. Therefore, you cannot unlink an object from its only linked folder or cabinet.

#### 2.29.1.4.3.1 HTTP Request method and query parameters

DELETE

#### 2.29.1.4.3.2 Request media types

N/A

#### 2.29.1.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.29.1.4.3.4 Request body

N/A

**2.29.1.4.3.5 Response headers**

N/A

**2.29.1.4.3.6 Response media types**

N/A

**2.29.1.4.3.7 Response status**

- 204 - Unlinked successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.29.1.4.3.8 Response body**

HTTP 204 No Content status upon a successful unlink operation. The response body contains no content.

## 2.29.2 Folder Child Links

The Folder Child Links collection resource represents a collection of all folder containment relationships for a folder. That is a set of links between a folder and its child objects.

**2.29.2.1 Resource relationships**

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder](#)” on page 233 and [About the Diagram on page 13](#).

**2.29.2.2 Feed**

Is this resource a feed? Yes.

- Feed ID: URI of the feed
- Feed Title: List of parent folders
- Feed Updated: r\_modify\_date of the object
- Supports POST: Yes
- Entry: “[Folder Child Links](#)” on page 277

- Entry ID: URI of the folder link
- Entry Title: Formatted string with both the child object ID and the parent object ID
- Entry Updated: `r_modify_date` of the child object.

### 2.29.2.3 Link relations

Link relation	Description	Resource reference
self	URI for Folder Child Links collection feed	<a href="#">"Folder Child Links" on page 277</a>
first, last, next, previous	Pagination links	<a href="#">"Folder Child Links" on page 277</a>

### 2.29.2.4 Operations

#### 2.29.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of child links for a given folder.
POST	Links an object to a given folder.

#### 2.29.2.4.2 Retrieve child links

Retrieve all child links for a given folder. That is a set of the links between the folder and all child objects under this folder.

##### 2.29.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	include-all-versions	Specifies whether or not to list all versions of the child object. <ul style="list-style-type: none"><li>• <code>true</code> - Returns all versions.</li><li>• <code>false</code> - Only returns the current version.</li></ul>	string	<code>true, false</code>	<code>false</code>

This operation also supports the following common query parameters:

- inline
- page
- items-per-page
- include-total
- sort
- filter
- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### **2.29.2.4.2.2 Request media types**

N/A

#### **2.29.2.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.29.2.4.2.4 Request body**

N/A

#### **2.29.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.29.2.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing

#### 2.29.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.29.2.4.2.8 Response body

XML or JSON representation of a collection child links for the given folder.

- The body contains a list of links between the given object and each of its parent folders.
- Each link contains the URI for the given folder and the URI for one of its child objects.
- Each entry contains the parent ID and the child ID.
- Each entry in the collection has a `self` link referring to the link resource.
- Pagination is supported.

#### 2.29.2.4.3 Link an object to a folder

Link a new object to a given folder or cabinet.

The first execution of link on an object defines the object's primary link, which is the place where the object is stored in the repository. Subsequent executions of link associate the object to other folders or cabinets. These links are called secondary links. This operation only allows you to create a new secondary link for an object. The existing folder links for this object do not change. For more information about how to create primary link during the object creation, see [“Folder Child Objects” on page 282](#).

To create a secondary link, you must have browse permission and the change location permission on the object. When the repository is running under folder security, you must also have the write permission on the folder or cabinet to which the object is linked.



**Note:** You cannot link a cabinet to a folder.

#### 2.29.2.4.3.1 HTTP Request method and query parameters

POST

#### 2.29.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.29.2.4.3.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.29.2.4.3.4 Request body

XML or JSON representation of the newly-created folder link between the given folder and the child object. The body contains the reference to the child object.

##### Example 2-108: Example of the Request body

```
{  
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
0900000c80000cc4"  
}
```



#### 2.29.2.4.3.5 Response headers

- Location
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.29.2.4.3.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.29.2.4.3.7 Response status

- 201 - Link created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.29.2.4.3.8 Response body

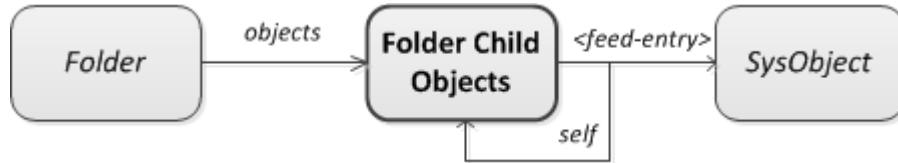
XML or JSON representation of newly-created link between the given folder and its new child object.

## 2.30 Folder Child Objects

The Folder Child Objects resource represents the collection of all the SysObject resources under a specified folder.

### 2.30.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder](#)” on page 233 and [About the Diagram](#) on page 13.

### 2.30.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the objects.
- Feed Title: List of folder child objects
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: “[SysObject](#)” on page 692
- Entry ID: URI of the object
- Entry Title: Object name
- Entry Summary: Object description
- Entry Updated: `r_modify_date` of the object

### 2.30.3 Link relations

Link relation	Description	Resource reference
self	The URI for folder child objects collection feed.	Folder child objects collection resource
first, last, next, previous	Pagination links.	Folder child objects collection resource

### 2.30.4 Operations

#### 2.30.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a list all objects under the specified folder.
POST	Creates a new object under the specified folder.
POST	Imports the object metadata and primary content.
POST	Copies an object to the specified folder.

#### 2.30.4.2 Get child objects from a folder

GET the collection of child objects under the given folder. You can specify any folder types, including `dm_folder`, `dm_cabinet` or a custom type.

##### 2.30.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	hide-shared-parent	<p>Specifies whether to show the shared parent object of the lightweight object.</p> <p> <b>Note:</b> Enabling hide-shared-parent may decrease performance.</p>	string	<i>true, false</i>	<i>false</i>
	include-all-versions	<p>Specifies whether or not to list all versions of the child object's link.</p> <ul style="list-style-type: none"> <li>• <i>true</i> - Returns all versions.</li> <li>• <i>false</i> - Only returns the current version</li> </ul>	string	<i>true, false</i>	<i>false</i>
	object-type	Specifies the exact sub type name to get from the collection of this type.	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	thumbnail	<p>Specifies whether this operation deletes a non-empty folder or not.</p> <ul style="list-style-type: none"><li>• <i>true</i> - Return the thumbnail link for each entry in the collection resource.</li><li>• <i>false</i> - Do not return the thumbnail link for each entry in the collection resource.</li></ul>	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	q	<p>Specifies the search criteria with a full-text expression in Simple Search Language.</p> <p>Parameter q must be encoded because it may contain non-English locale characters.</p> <p> <b>Note:</b> International characters that are used in this query parameter must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.</p>	string	N/A	No search criteria is used for the search.

This method also supports the following common query parameters:

- `inline`
- `sort`

The default sort order is by creation date, descending (most recent to oldest).

- page
- items-per-page
- include-total
- view
- links
- filter
- q

The q parameter is supported for full text searching with a subset of the Simple Search Language, however parentheses are not supported.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### **2.30.4.2.2 Request media types**

N/A

#### **2.30.4.2.3 Request headers**

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.30.4.2.4 Request body**

N/A

#### **2.30.4.2.5 Response headers**

- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.30.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.30.4.2.8 Response body

XML or JSON representation of the collection of child objects under the specified folder.

- The body contains a list of `SysObject` resources, each of which is filed under the specified folder.
- Each object may contain all properties of the object, depending on the setting of the query parameter `view`.
- Each object contains links specific to the type of the object. See [“Link relations” on page 693](#) in `SysObject` resource, which provides information on what links are available for a certain type of object.
- The returned child objects collection only contains those that you have access to.

### 2.30.4.3 Create an object in a folder

Create a child object under the specified folder. You can set the properties when creating the object. The object can be the type of `dm_SysObject` or its subtype. By default, the object type is `dm_SysObject`.

#### 2.30.4.3.1 HTTP Request method and query parameters

POST

When storing content to Centera storage, the `content-length` query parameter must be provided. When providing the value for `content-length`, you must provide an accurate byte count of the content to be uploaded. For other storage mediums, such as harddisk, the `content-length` parameter is not required and is optional. However, even when not required, if this parameter is provided, it must be accurate.



**Note:** When creating a document that contains no content, neither `content-length` or `content charset` is used. These two variables are only used when your documents contain content.

#### 2.30.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.30.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.3.4 Request body

XML or JSON representation of the object to create.

- The object type is specified in the `r_object_type` property of the request body. The `r_object_type` property is optional. If this property is not specified in a client request, it is set to `dm_folder` by default.
- Documentum CM Server validates the `object_name` property in the request. This property is mandatory for some types, for example, `dm_folder`.
- Documentum CM Server validates the constraints and permissions on setting the other properties (including read-only properties).

#### 2.30.4.3.5 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.30.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.30.4.3.8 Response body

XML or JSON representation of the SysObject resource created under the specified folder.

- Each object contains all properties of the object.
- Each object contains links specific to the type of the object. For information on links that are available for a certain type of object, see ["Link relations" on page 693](#) in SysObject resource.

### 2.30.4.4 Import SysObject metadata and contents

Create a document and its contents under a specified folder in a single request.

#### 2.30.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	content-count	Specifies how many pieces of content are submitted. The metadata part is excluded from the count.  For more information, see <a href="#">content-count</a> .	integer	>=0	0

HTTP method	Query parameters	Description	Data type	Value range	Default value
	all-primary	Specifies whether to import all contents using the same type as the primary content for different pages (page 0, 1, 2...), or one primary content, and others as renditions for the first page, which is page 0	string	true or <i>false</i>	<i>true</i>
	format	<p>Specifies the <code>dm_format</code> name for contents.</p> <p>When more than one type of contents are being imported, the <code>format</code> parameter can be used to define the format for each piece of content by using a comma separated string.</p> <p>For example, <code>format=crtext,html,pub_html</code></p> <p>For more information, see <a href="#">format</a>.</p>	string	Comma separated string	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	modifier	<p>Specifies the modifier of the content. For the primary content, the modifier is ignored.</p> <p>For example, when you import with the all-primary parameter set to <i>false</i>, the value of modifier can be: <code>modifier=, mod1,mod2, mod3</code>. That means the first (or primary) contents has no modifier, and the second rendition's modifier is mod1, the third rendition's modifier is mod2 and so on.</p> <p>For more information, see <a href="#">modifier</a>.</p>	string	Comma separated string	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-length	<p>Specifies the byte count of the content to be uploaded. This parameter is optional.</p> <p>A comma separated list can be used to define multiple values for this parameter. For example: content-length=1024,2048,,4096. The blank value shown here is 0.</p> <p>For more information, see <a href="#">content-length</a>.</p>	string	Comma separated string	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	content-charset	<p>This parameter is used to sanitize content and it defines how the server parses the content while sanitizing. Multiple values for this parameter can be set using a comma separated list. For example: content-charset= UTF-8,GBK, ISO-8859-1.</p> <p>For more information, see <a href="#">content-charset</a>.</p>	string	Comma separated string	null



**Note:** Here are some additional details about the variables listed above:

- **format**
  - The format list must match the content sequence. When you don't want to specify the format for a piece of content you can leave it blank in the comma separated list but you must still include the comma as a placeholder. For example: format=crtext,,html.
  - When the format for contents is not defined, the server uses the content media type to select the format.
  - When the all-primary parameter is *true*, all primary content types must have the same format. In this case, when you define more than one format, they must be the same. For example: format=crtext,crtext,crtext. You can also define one format for all of the primary content: format=crtext.
- **modifier**

Setting the all-primary parameter to true , or when content-count<=1 causes all modifiers to be ignored.
- **content-charset**

- When the metadata has a valid charset, then it is used to sanitize the content.
  - When the content-charset is not provided and the content is in HTML format, the server tries to get a value for the content-charset parameter from the metadata information of the content itself.
  - The configuration default charset is used when the content-charset parameter and the metadata information are not found
  - When the charset and the content do not match, the uploaded content may have an incorrectly encoded value.
  - The configuration default charset is used when the content-charset parameter and the metadata information are not found
- content-length
    - When the all-primary parameter is set to *false*, the first value of in the content-length parameter is used for the primary piece of content, the other values in the comma separated string are ignored.
    - When storing content to Centera storage, the content-length parameter is mandatory and must be provided. For other storage mediums, such as hard disk, the content-length parameter is optional.
    - In all cases, including when the content-length parameter is not required, if a value for it is set, the value must be an accurate byte count of the contents to be uploaded.
    - When the all-primary parameter is set to *false*, the first value in the comma separated content-length string parameter is used for the primary piece of content, but the other values in the comma separated string are ignored.
  - content-count
    - When you import more than one piece of contents, the REST server must know how many content pieces there are in total. The total number of content pieces cannot be obtained until the end of the multipart stream. Therefore, the client must specify this number in the value of the content-count parameter.
    - When the value set in the content-count parameter is greater than the actual number of content pieces, an exception is thrown because the system expects more content pieces.
    - When the value set in the content-count parameter is less than the actual number of content pieces, all content pieces that exist in addition to the number of pieces set in the content-count parameter are ignored.



### Caution

Ensure that the value provided by the content-count is accurate.

#### 2.30.4.4.2 Request media types

- multipart/form-data
- multipart/mixed

#### 2.30.4.4.3 Request headers

- Accept
- Authorization
- Content-Type (The Content-Type header must be either multipart/form-data or multipart/mixed, which notifies the server this request is a multipart request.)

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.4.4 Request body

The request body is a multipart message that must conform to RFC 2046. A multipart/form-data or multipart/mixed message contains a series of parts. This series must conform to the following rules:

- The series must contain two parts.
- The first part in the series must contain the metadata for the document to be created.

Content-Type for the first part must be one of the following values:

- application/vnd.emc.documentum+json (default)
- application/vnd.emc.documentum+xml

When the REST client sets Content-Type to a value that is invalid, the server uses application/vnd.emc.documentum+json to overwrite the invalid value.

- The second part in the series must be the binary data of the primary content. Content-Type for the second part must be the MIME type for the primary content.
- Each part in the series must contain a Content-Disposition header. The header's disposition type must be form-data. The disposition must contain a parameter titled name, which specifies the name of this part.

#### ➡ Example 2-109: XML request - multipart/form-data or multipart/mixed

```
--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+xml

<object>
  <properties>
    <object_name>xyz</object_name>
  </properties>
</object>
```

```
--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the first content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the second content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the third content part

--314159265358979--
```



#### ➡ Example 2-110: JSON request - multipart/form-data or multipart/mixed

```
--314159265358979
Content-Disposition: form-data; name=metadata
Content-Type: application/vnd.emc.documentum+json
{
  "properties": {
    "object_name": "xyz"
  }
}

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the first content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the second content part

--314159265358979
Content-Disposition: form-data; name=binary
Content-Type: text/plain

This is the third content part

--314159265358979--
```



#### 2.30.4.4.5 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.4.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.30.4.4.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.30.4.4.8 Response body

XML or JSON representation of the SysObject resource created under the specified folder.

- Each object contains all properties of the object.
- Each object contains links specific to the type of the object. See [“Link relations” on page 693](#) in SysObject resource, which provides information on what links are available for a certain type of object.

 **Example 2-111: XML response - application/vnd.emc.documentum+xml or “application/xml”**

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://identifiers.emc.com/vocab/documentum"
    definition="{typeResourceUri}"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_document">
    <properties>
        <object_name>xyz</object_name>
        <r_object_id>090020808001398e</r_object_id>
        ...
    </properties>
    <links>
        <link rel="self" href="{documentResourceUri}" />
        ...
    </links>
</document>
```

```
</links>
</document>
```



► **Example 2-112: JSON response - application/vnd.emc.documentum+xml or “application/xml”**

```
{
  "name": "document",
  "type": "dm_document",
  "definition": "<documentResourceUri>",
  "properties": {
    "object_name": "xyz",
    "r_object_id": "090020808001398e",
    ...
  }
  "links": [
    { "rel": "self", "href": "<documentResourceUri>" },
    ...
  ]
}
```



#### 2.30.4.5 Copy an object to a folder

Copy an object to a folder. The type of the object must be `dm_SysObject` or its subtype. Properties of the object can be customized during the copy operation.



**Note:** When you copy a folder, whether and how to copy the objects under the folder depends on the variables set in the message body.

For more information, see “[Request body](#)” on page 300.

##### 2.30.4.5.1 HTTP Request method and query parameters

POST

##### 2.30.4.5.2 Request media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

#### 2.30.4.5.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.5.4 Request body

The request body contains the link to the object to copy, and the following variables.

Variable	Description	Data type	Default value
deep-copy	Specifies whether or not to copy the objects under the folder: <ul style="list-style-type: none"><li>• <i>true</i> - copy the objects under the folder</li><li>• <i>false</i> - copy the folder only.</li></ul>	boolean	<i>true</i>
replicate-vd-children	When the value of the deep-copy parameter is <i>true</i> and you copy a folder, copying a virtual document that is within that folder will cause the children of that virtual document to be copied as well.  When the value of the deep-copy parameter is <i>false</i> and you copy a folder, the children of that virtual document are only referenced.	boolean	<i>true</i>

Variable	Description	Data type	Default value
retain-acl	<p>This parameter determines whether to apply a source object's ACL to the target object while copying an object.</p> <p>When this parameter is set to <i>true</i>:</p> <ul style="list-style-type: none"> <li>When the source is an object or the source is a folder that has the deep-copy parameter set to <i>false</i>, this parameter causes the source ACL to be applied only to the newly copied object or folder.</li> <li>When the source is a folder with the deep-copy parameter set to <i>true</i>, the ACL of all the source objects within that folder are applied to their related newly copied objects.</li> </ul> <p>When the retain-acl parameter is <i>false</i>, no ACL is applied to the target of the copy procedure.</p>	boolean	<i>false</i>
include-broken	Specifies whether or not to include binding broken components while copying a virtual document.	boolean	<i>false</i>
binding-label	Specifies the late binding version while copying a virtual document.	string	CURRENT

#### ➡ Example 2-113: JSON request payload

```
{
  "href": "http://localhost/dctm-rest/repositories/acme/objects/xxxxx",
  "deep-copy": true,
  "replicate-vd-children": true,
  "retain-acl": true
  "include-broken": false
  "binding-label": CURRENT
}
```



#### ➡ Example 2-114: XML request payload

```
<dm:object href="http://localhost/dctm-rest/repositories/acme/objects/xxxxx"
  deep-copy="true" replicate-vd-children="true" retain-acl="true"
  include-broken="false" binding-label="CURRENT" />
```



#### 2.30.4.5.5 Response headers

- Location
- Content-Length
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.30.4.5.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.30.4.5.7 Response status

- 200 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.30.4.5.8 Response body

XML or JSON representation of the object copied to this folder.

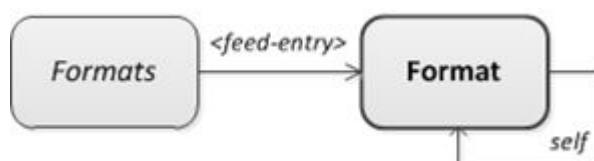
## 2.31 Format (s)

### 2.31.1 Format

The Format resource represents a `dm_format` instance stored in a repository.

#### 2.31.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Formats” on page 305 and About the Diagram on page 13.

### 2.31.1.2 Feed

No feed support.

### 2.31.1.3 Link relations

Link relation	Description	Resource reference
self	This format	"Format" on page 302

### 2.31.1.4 Operations

#### 2.31.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the information about a format.

#### 2.31.1.4.2 Get a format

Retrieves the information about a format by a specified format name.

##### 2.31.1.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- view
- links
- sort
- inline
- filter
- page
- items-per-page
- include-total

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.31.1.4.2.2 Request media types

N/A

#### 2.31.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.31.1.4.2.4 Request body

N/A

#### 2.31.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.31.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.31.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.31.1.4.2.8 Response body

XML or JSON representation of the format instance.



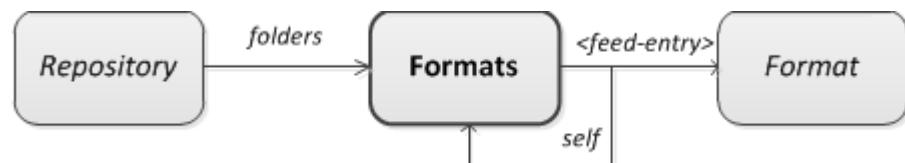
**Note:** The format name in the URL is encoded.

## 2.31.2 Formats

The Formats resource represents a collection of `dm_format` instances stored in a repository.

### 2.31.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.31.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of formats
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Format” on page 302
- Entry ID: URI of the format
- Entry Title: Format name
- Entry Summary: Format description
- Entry Updated: `r_modify_date` of the format

### 2.31.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of formats.	“Formats” on page 305
first, last, next, previous	Pagination links.	“Formats” on page 305

## 2.31.2.4 Operations

### 2.31.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the information about a collection of formats.

### 2.31.2.4.2 Get formats

Retrieve the information about a collection of formats.

#### 2.31.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- view
- items-per-page
- page
- include-total
- sort
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.31.2.4.2.2 Request media types

N/A

#### 2.31.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.31.2.4.2.4 Request body

N/A

#### 2.31.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.31.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.31.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.31.2.4.2.8 Response body

XML or JSON representation of the collection of formats.

- The body contains a list of the `dm_format` instances (or subtypes of `dm_format`).
- Each object may contain all or a set of properties of the format, depending on the setting of the query parameter `view`.
- The returned child objects collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order by format name.
- The format name in the URL is encoded.

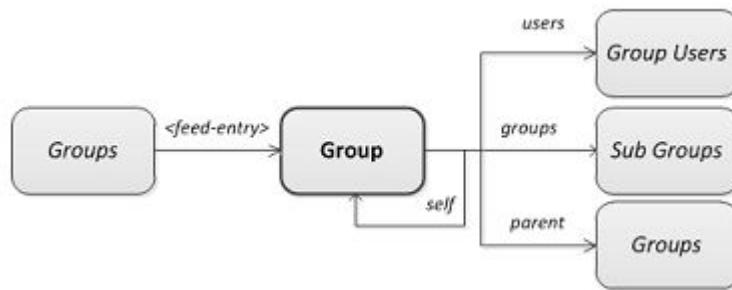
## 2.32 Group (s)

### 2.32.1 Group

The Group resource represents a group in a repository.

#### 2.32.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Groups” on page 317 and About the Diagram on page 13.

#### 2.32.1.2 Feed

No feed support.

#### 2.32.1.3 Link relations

Link relation	Description	Resource reference
self	This Group resource.	<a href="#">“Group” on page 308</a>
edit	This Group resource. Only sysAdmin, SuperUser or group owner can view this link	<a href="#">“Group” on page 308</a>
delete	This Group resource. Only sysAdmin, SuperUser or group owner can view this link	<a href="#">“Group” on page 308</a>
users [1]	Users that directly belong to this group.	<a href="#">“Sub Groups” on page 677</a>
groups [1]	Sub groups that directly belong to this group.	<a href="#">“Sub Groups” on page 677</a>
parent	Groups that this group belongs to.	<a href="#">“Groups” on page 317</a>

Link relation	Description	Resource reference
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.32.1.4 Operations

### 2.32.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieve a group instance
POST	Update a group
DELETE	Delete a group

### 2.32.1.4.2 Get a group

Retrieve all attributes of a group.

#### 2.32.1.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- view
- links
- inline
- filter
- sort
- page
- items-per-page
- include-total

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.32.1.4.2.2 Request media types

N/A

#### 2.32.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.32.1.4.2.4 Request body

N/A

#### 2.32.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.32.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.32.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.32.1.4.2.8 Response body

XML or JSON representation of the group. The information about the parent groups and sub groups are presented in link relations instead of properties.



**Note:** The group name in the URL is encoded.



##### Example 2-115: XML response

```
<?xml version='1.0' encoding='UTF-8' ?>
<group xsi:type="dm_group" definition="http://localhost:8080/dctm-rest/
repositories/
```

```

REPO/types/dm_group.xml">
<properties>
    <group_name>admingroup</group_name>
    <group_address />
    <users_names>
        <item>REPO_ADMIN</item>
        <item>Administrator</item>
    </users_names>
    <owner_name>Administrator</owner_name>
    <is_private>false</is_private>
    <description />
    <globally_managed>false</globally_managed>
    <r_modify_date>2015-03-03T02:20:31.000+00:00</r_modify_date>
    <alias_set_id>0000000000000000</alias_set_id>
    <group_source />
    <group_class>group</group_class>
    <group_admin />
    <r_has_events>false</r_has_events>
    <is_dynamic>false</is_dynamic>
    <is_dynamic_default>false</is_dynamic_default>
    <group_global_unique_id>REPO:admingroup</group_global_unique_id>
    <group_native_room_id>0000000000000000</group_native_room_id>
    <group_directory_id>0000000000000000</group_directory_id>
    <group_display_name>admingroup</group_display_name>
    <is_protected>false</is_protected>
    <is_module_only>false</is_module_only>
    <i_is_replica>false</i_is_replica>
    <i_vstamp>0</i_vstamp>
    <r_object_id>120000580000129</r_object_id>
</properties>
<links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
        admingroup.xml" />
    <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
        admingroup.xml" />
    <link rel="delete" href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
        admingroup.xml" />
    <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/REPO/
        groups.xml?group-name=admingroup" />
    <link rel="http://identifiers.emc.com/linkrel/groups"
        href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
        admingroup/
            groups.xml" />
    <link rel="http://identifiers.emc.com/linkrel/users"
        href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
        admingroup/
            users.xml" />
</links>
</group>

```



#### Example 2-116: JSON response

```
{
    "name": "group",
    "type": "dm_group",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
```

```
dm_group",
  "properties": {
    "group_name": "admingroup",
    "group_address": "",
    "users_names": ["REPO_ADMIN", "Administrator"],
    "owner_name": "Administrator",
    "is_private": false,
    "description": "",
    "globally_managed": false,
    "r_modify_date": "2015-03-03T02:20:31.000+00:00",
    "alias_set_id": "0000000000000000",
    "group_source": "",
    "group_class": "group",
    "group_admin": "",
    "r_has_events": false,
    "is_dynamic": false,
    "is_dynamic_default": false,
    "group_global_unique_id": "REPO:admingroup",
    "group_native_room_id": "0000000000000000",
    "group_directory_id": "0000000000000000",
    "group_display_name": "admingroup",
    "is_protected": false,
    "is_module_only": false,
    "i_is_replica": false,
    "i_vstamp": 0,
    "r_object_id": "1200000580000129"
  },
  "links": [
    {"rel": "self",
     "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/admingroup"},
    {"rel": "edit",
     "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/admingroup"},
    {"rel": "delete",
     "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/admingroup"},
    {"rel": "parent",
     "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups?group-name=admingroup"},
    {"rel": "http://identifiers.emc.com/linkrel/groups",
     "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/admingroup/groups"},
    {"rel": "http://identifiers.emc.com/linkrel/users",
     "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/admingroup/users"}
  ]
}
```



#### 2.32.1.4.3 Update a group

Update the attributes of a group. Only the SysAdmin, SuperUser, or group owner can update the group.

##### 2.32.1.4.3.1 HTTP Request method and query parameters

POST

##### 2.32.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

##### 2.32.1.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.32.1.4.3.4 Request body

###### ➡ Example 2-117: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:dm_group xsi:type="dm_group" definition="http://localhost:8080/dctm-rest/
repositories/REPO/types/dm_group.xml">
    <dm:properties>
        <dm:owner_name>Administrator</dm:owner_name>
        .....
    </dm:properties>
</dm:dm_group>
```



###### ➡ Example 2-118: JSON request

```
{
    properties:{
        "owner_name": "Administrator",
        ...
    }
}
```



#### 2.32.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.32.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.32.1.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.32.1.4.3.8 Response body

##### Example 2-119: XML response

```
<?xml version='1.0' encoding='UTF-8' ?>
<group xsi:type="dm_group" definition="http://localhost:8080/dctm-rest/
repositories/
REPO/types/dm_group.xml">
<properties>
    <group_name>admingroup</group_name>
    <group_address />
    <users_names>
        <item>REPO_ADMIN</item>
        <item>Administrator</item>
    </users_names>
    <owner_name>Administrator</owner_name>
    <is_private>false</is_private>
    <description />
    <globally_managed>false</globally_managed>
    <r_modify_date>2015-03-03T02:20:31.000+00:00</r_modify_date>
    <alias_set_id>0000000000000000</alias_set_id>
    <group_source />
    <group_class>group</group_class>
    <group_admin />
    <r_has_events>false</r_has_events>
    <is_dynamic>false</is_dynamic>
    <is_dynamic_default>false</is_dynamic_default>
    <group_global_unique_id>REPO:admingroup</group_global_unique_id>
    <group_native_room_id>0000000000000000</group_native_room_id>
    <group_directory_id>0000000000000000</group_directory_id>
    <group_display_name>admingroup</group_display_name>
    <is_protected>false</is_protected>
    <is_module_only>false</is_module_only>
    <i_is_replica>false</i_is_replica>
    <i_vstamp>0</i_vstamp>
    <r_object_id>1200000580000129</r_object_id>
</properties>
```

```

<links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
groups/
        admingroup.xml" />
    <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/
groups/
        admingroup.xml" />
    <link rel="delete" href="http://localhost:8080/dctm-rest/repositories/REPO/
groups/
        admingroup.xml" />
    <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/REPO/
groups.xml?group-name=admingroup" />
    <link rel="http://identifiers.emc.com/linkrel/groups"
        href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup/
        groups.xml" />
    <link rel="http://identifiers.emc.com/linkrel/users"
        href="http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup/
        users.xml" />
</links>
</group>
```



#### Example 2-120: JSON response

```
{
    "name": "group",
    "type": "dm_group",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_group",
    "properties": {
        "group_name": "admingroup",
        "group_address": "",
        "users_names": ["REPO_ADMIN",
        "Administrator"],
        "owner_name": "Administrator",
        "is_private": false,
        "description": "",
        "globally_managed": false,
        "r_modify_date": "2015-03-03T02:20:31.000+00:00",
        "alias_set_id": "0000000000000000",
        "group_source": "",
        "group_class": "group",
        "group_admin": "",
        "r_has_events": false,
        "is_dynamic": false,
        "is_dynamic_default": false,
        "group_global_unique_id": "REPO:admingroup",
        "group_native_room_id": "0000000000000000",
        "group_directory_id": "0000000000000000",
        "group_display_name": "admingroup",
        "is_protected": false,
        "is_module_only": false,
        "i_is_replica": false,
        "i_vstamp": 0,
        "r_object_id": "1200000580000129"
    },
    "links": [
        {"rel": "self",
```

```
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup"},  
        {"rel": "edit",  
         "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup"},  
        {"rel": "delete",  
         "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup"},  
        {"rel": "parent",  
         "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups?
group-name=admingroup"},  
        {"rel": "http://identifiers.emc.com/linkrel/groups",  
         "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup/groups"},  
        {"rel": "http://identifiers.emc.com/linkrel/users",  
         "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/
admingroup/users"}  
    ]  
}
```



#### 2.32.1.4.4 Delete a group

Delete a group. Only the SysAdmin, SuperUser, or group owner can delete a group.

##### 2.32.1.4.4.1 HTTP Request method and query parameters

DELETE

##### 2.32.1.4.4.2 Request media types

N/A

##### 2.32.1.4.4.3 Request headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.32.1.4.4.4 Request body

N/A

##### 2.32.1.4.4.5 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.32.1.4.4.6 Response media types

N/A

#### 2.32.1.4.4.7 Response status

- 204 - Deleted successfully
- 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.32.1.4.4.8 Response body

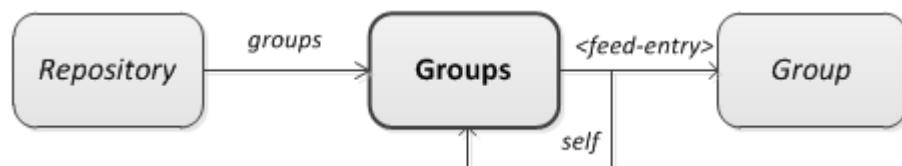
N/A

## 2.32.2 Groups

The Groups resource represents a collection of group instances in a repository.

### 2.32.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

### 2.32.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of groups
- Supports POST: No
- Entry: “[Group](#)” on page 308
- Entry ID: URI of the group
- Entry Title: Group name
- Entry Summary: Group description
- Entry Updated: `r_modify_date` of the group

### 2.32.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of groups.	<a href="#">“Groups” on page 317</a>
first, last, next, previous	Pagination links.	<a href="#">“Groups” on page 317</a>

### 2.32.2.4 Operations

#### 2.32.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the information about a collection of groups.

#### 2.32.2.4.2 Get groups

Retrieve the information about a collection of groups in a repository.

##### 2.32.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Default value
GET	user-name	Only groups that contain a user (a dm_user instance) whose user-name property equals to the specified string are returned in the result.	string	null
	group-name	Only groups that contain a sub group (a dm_group instance) whose group-name property equals to the specified string are returned in the result.	string	null



#### Caution

You cannot use the username and groupname parameters simultaneously.

This method also supports the following common query parameters:

- inline
- view
- items-per-page
- page
- include-total
- sort
- recursive
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.



**Note:** International characters that are used in the preceding query parameters must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.

#### 2.32.2.4.2.2 Request media types

N/A

#### 2.32.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.32.2.4.2.4 Request body

N/A

#### 2.32.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.32.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.32.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.32.2.4.2.8 Response body

XML or JSON representation of the collection of groups.

- The body contains a list of the dm\_group instances (or subtypes of dm\_group).
- Each object may contain all or a set of properties of the group, depending on the setting of the query parameter view.
- The returned child objects collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order by group name.
- The group name in the URL is encoded.

#### 2.32.2.4.3 Create a group

Create a group in the repository. The SysAdmin, SuperUser or any other user with “Create group” permissions can create a group.

##### 2.32.2.4.3.1 HTTP Request method and query parameters

POST

#### 2.32.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.32.2.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.32.2.4.3.4 Request body

The group\_name property is mandatory.

##### ➡ Example 2-121: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<dm_group xsi:type="dm_group" definition="http://localhost:8080/dctm-rest/
repositories/
REPO/types/dm_group">
<properties>
    <group_name>group1</group_name>
    ...
</properties>
</dm_group>
```



##### ➡ Example 2-122: JSON request

```
{
    "properties" : {
        "group_name" : "group1",
        ...
    }
}
```



#### 2.32.2.4.3.5 Response headers

- Location

The URL of the created group.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.32.2.4.3.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json

#### 2.32.2.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.32.2.4.3.8 Response body

N/A

### 2.33 Group User (s)

#### 2.33.1 Group User

The Group User resource represents the relationship between a group and one of its user members.

##### 2.33.1.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* ["Group Users" on page 324](#) and [About the Diagram on page 13](#).

### 2.33.1.2 Feed

No feed support.

### 2.33.1.3 Link relations

N/A

### 2.33.1.4 Operations

#### 2.33.1.4.1 Supported HTTP methods

Method	Description
DELETE	Remove the relation between a group and its user member

#### 2.33.1.4.2 Delete a user relationship

Remove the relation between a group and its user member.

##### 2.33.1.4.2.1 HTTP Request method and query parameters

DELETE

##### 2.33.1.4.2.2 Request media types

N/A

##### 2.33.1.4.2.3 Request headers

- Accept
- Content-Type
- Authentication

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.33.1.4.2.4 Request body

###### ➡ Example 2-123: Request - Delete a user relationship

```
DELETE http://localhost:8080/dctm-rest/repositories/acme/groups/  
mydocument/users/myuser
```



#### 2.33.1.4.2.5 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.33.1.4.2.6 Response media types

N/A

#### 2.33.1.4.2.7 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.33.1.4.2.8 Response body

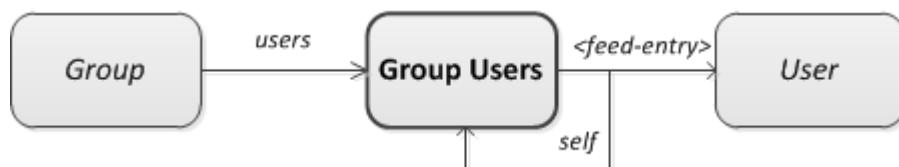
HTTP 204 No Content status upon a successful delete operation.

## 2.33.2 Group Users

The Group Users resource represents a collection of users that belong to a specified group.

### 2.33.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Group](#)” on page 308 and [About the Diagram on page 13](#).

### 2.33.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of users
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “[User](#)” on page 766
- Entry ID: URI of the user

- Entry Title: User name
- Entry Summary: User description
- Entry Updated: r\_modify\_date of the user

### 2.33.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of users.	"Group Users" on page 324
first, last, next, previous	Pagination links.	"Group Users" on page 324

### 2.33.2.4 Operations

#### 2.33.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the information about a collection of users.

#### 2.33.2.4.2 Get group users

Retrieve the information about a collection of users that belong to a given group.

##### 2.33.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- view
- items-per-page
- page
- include-total
- sort
- recursive
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### **2.33.2.4.2.2 Request media types**

N/A

#### **2.33.2.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.33.2.4.2.4 Request body**

N/A

#### **2.33.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.33.2.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.33.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.33.2.4.2.8 Response body**

XML or JSON representation of the collection of users.

- The body contains a list of the `dm_user` instances (or subtypes of `dm_user`).
- Each object may contain all or a set of properties of the user, depending on the setting of the query parameter `view`.
- The returned child objects collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order by user name.

#### 2.33.2.4.3 Add a user member

Add an existing group into a group as a user member.

##### 2.33.2.4.3.1 HTTP Request method and query parameters

POST

- href

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.33.2.4.3.2 Request media types

- application/atom+xml
- application/vnd.emc.documentum+json

##### 2.33.2.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.33.2.4.3.4 Request body

The href variable, which specifies the user resource, must be supplied. This variable is a string and has a <null> default value.

###### ➡ Example 2-124: XML request

```
<dm:user href="http://localhost/dctm-rest/repositories/acme/users/xxxxx" />
```



###### ➡ Example 2-125: JSON request

```
{  
    "href": "http://localhost/dctm-rest/repositories/acme/users/xxxxx"  
}
```



#### 2.33.2.4.3.5 Response headers

- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.33.2.4.3.6 Response media types

N/A

#### 2.33.2.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.33.2.4.3.8 Response body

N/A

## 2.34 Get Combined Groups and Members

Retrieves groups and members for the requested groups or roles. It is a generic API whose response is the combination of two existing API response such as group-member-groups and group-member-users.

The response contains the name, address, r\_is\_group, and group\_class.

### 2.34.1 URI

- Item: Base URI
- Value: /repositories/{repositoryName}/groups/{groupName}/members

### 2.34.2 Navigation

Groups

### 2.34.3 Link relations

`http://identifiers.emc.com/linkrel/members`

### 2.34.4 HTTP Request method

GET

### 2.34.5 Query parameters

Supports the following common query parameters:

- inline
- links
- view (only default is supported)
- page
- items-per-page

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

### 2.34.6 Request media types

`application/vnd.emc.documentum+json`

### 2.34.7 Request headers

Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.34.8 Request body

N/A

### 2.34.9 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.34.10 Response media types

application/vnd.emc.documentum+json

### 2.34.11 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.34.12 Response body

JSON representation of the combined groups and members.

➡ **Example 2-126: JSON response**

```
{  
    "id": "http://localhost:8080/dctm-rest/repositories/d2repo/groups/  
bws_0b0003e98000a26e_admin/members",  
    "title": "",  
    "author": [  
        {  
            "name": "Open Text Documentum"  
        }  
    ],  
    "updated": "2025-07-16T06:35:59.197+00:00",  
    "page": 1,  
    "items-per-page": 100,  
    "links": [  
        {  
            "rel": "self",  
            "href": "http://localhost:8080/dctm-rest/repositories/d2repo/  
groups/bws_0b0003e98000a26e_admin/members?  
inline=true&links=true&view=:default&page=1&items-per-page=100"  
        }  
    ],  
    "entries": [  
        {  
            "id": "admingroup",  
            "title": "admingroup",  
            "content": {  
                "name": "member",  
                "properties": {  
                    "name": "admingroup",  
                    "r_is_group": true  
                }  
            }  
        }  
    ]  
}
```

```

        },
        "links": [
            {
                "rel": "self",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admingroup"
            },
            {
                "rel": "http://identifiers.emc.com/linkrel/groups",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admingroup/groups"
            }
        ]
    }
}

{
    "id": "c2admin",
    "title": "c2admin",
    "content": {
        "name": "member",
        "properties": {
            "name": "c2admin",
            "email_address": "c2admin",
            "r_is_group": false
        },
        "links": [
            {
                "rel": "self",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/users/c2admin"
            },
            {
                "rel": "http://identifiers.emc.com/linkrel/users",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/c2admin/users"
            }
        ]
    }
},
{
    "id": "d2-base_users",
    "title": "d2-base_users",
    "content": {
        "name": "member",
        "properties": {
            "name": "d2-base_users",
            "email_address": "d2-base_users",
            "r_is_group": true
        },
        "links": [
            {
                "rel": "self",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/d2-base_users"
            },
            {
                "rel": "http://identifiers.emc.com/linkrel/groups",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/d2-base_users/groups"
            }
        ]
    }
}

```

```
        },
        {
            "id": "d2_privileged_users",
            "title": "d2_privileged_users",
            "content": {
                "name": "member",
                "properties": {
                    "name": "d2_privileged_users",
                    "r_is_group": true
                },
                "links": [
                    {
                        "rel": "self",
                        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/d2_privileged_users"
                    },
                    {
                        "rel": "http://identifiers.emc.com/linkrel/groups",
                        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/d2_privileged_users/groups"
                    }
                ]
            }
        },
        {
            "id": "dmadmin",
            "title": "dmadmin",
            "content": {
                "name": "member",
                "properties": {
                    "name": "dmadmin",
                    "email_address": "dmadmin@d2vm.com",
                    "r_is_group": false
                },
                "links": [
                    {
                        "rel": "self",
                        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/users/dmadmin"
                    },
                    {
                        "rel": "http://identifiers.emc.com/linkrel/users",
                        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/dmadmin/users"
                    }
                ]
            }
        },
        {
            "id": "otadmin",
            "title": "otadmin",
            "content": {
                "name": "member",
                "properties": {
                    "name": "otadmin",
                    "r_is_group": false
                },
                "links": [
                    {
                        "rel": "self",
                        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/users/otadmin"
                    }
                ]
            }
        }
    ]
}
```

```

        },
        {
            "rel": "http://identifiers.emc.com/linkrel/users",
            "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/otadmin/users"
        }
    ]
}
{
    "id": "test_otds_user",
    "title": "test_otds_user",
    "content": {
        "name": "member",
        "properties": {
            "name": "test_otds_user",
            "email_address": "test@d2vm.com",
            "r_is_group": false
        },
        "links": [
            {
                "rel": "self",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/users/test_otds_user"
            },
            {
                "rel": "http://identifiers.emc.com/linkrel/users",
                "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/test_otds_user/users"
            }
        ]
    }
}
]
}

```



## 2.35 Home Document

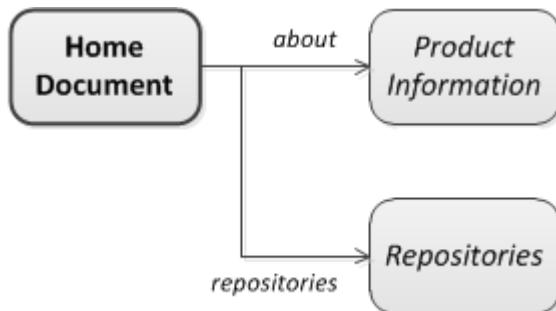
The Home Document collection resource provides an entry point for available resources. The design of Home Document adheres to the following RFC documents:

[Home Documents for HTTP Services: XML Syntax](#) on IETF Tools website (XML representation)

[Home Documents for HTTP APIs](#) on IETF Tools website (JSON representation)

### 2.35.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* [About the Diagram on page 13](#)

### 2.35.2 Feed

No feed support.

### 2.35.3 Link relations

Link relation	Description	Resource reference
repositories [1]	Link to access any available repositories	<a href="#">"Repositories" on page 573</a>
about	Link to production information	<a href="#">"Product Information" on page 517</a>
search	Link that can be used to perform a multiple repositories search	<a href="#">"Multiple Repository Search" on page 432</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
<http://identifiers.emc.com/linkrel/>.

### 2.35.4 Operations

#### 2.35.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the home document.

#### 2.35.4.2 Get the home document

Retrieves the home document which provides an entry point for available resources. The URI of the Home Document resource is constant, for example `/services`. Your REST clients can use the `/services` URI to access Foundation REST API.

For example, when the context root of the REST server is deployed as `http://localhost:8080/dctm-rest`, then the Home Document's URI would be `http://localhost:8080/dctm-rest/services`.

##### 2.35.4.2.1 HTTP Request method and query parameters

GET

##### 2.35.4.2.2 Request media types

N/A

##### 2.35.4.2.3 Request headers

- Accept

This resource does not accept `application/vnd.emc.documentum+json` or `application/vnd.emc.documentum+xml`. Instead, it accepts `application/home+json` and `application/home+xml`.



**Note:** The authorization header is not needed when a client tries to get this resource.

For more information about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.35.4.2.4 Request body

N/A

#### 2.35.4.2.5 Response headers

- Content-Type
- Content-Length

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.35.4.2.6 Response media types

- application/home+xml
- application/home+json

#### 2.35.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.35.4.2.8 Response body

The following code sample illustrates the Response body for this resource.

##### ➡ Example 2-127: XML response

```
<resources>
  <resource rel="http://identifiers.emc.com/linkrel/repositories">
    <link href=
      "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories" />
    <hints>
      <allow>
        <i>GET</i>
      </allow>
      <representations>
        <i>application/xml</i>
        <i>application/json</i>
        <i>application/atom+xml</i>
        <i>application/vnd.emc.documentum+json</i>
      </representations>
    </hints>
  </resource>
  <resource rel="about">
    <link href= "http://core-rs-demo.lss.emc.com:8080/dctm-rest/product-
info" />
    <hints>
      <allow>
        <i>GET</i>
      </allow>
      <representations>
        <i>application/xml</i>
        <i>application/json</i>
        <i>application/vnd.emc.documentum+xml</i>
        <i>application/vnd.emc.documentum+json</i>
      </representations>
    </hints>
  </resource>
</resources>
```

```

    </hints>
</resource>
</resources>

```



### ➤ Example 2-128: JSON response

```

{
  "resources": {
    "http://identifiers.emc.com/linkrel/repositories": {
      "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/
repositories.json",
      "hints": {
        "allow": [
          "GET"
        ],
        "representations": [
          "application/xml",
          "application/json",
          "application/atom+xml",
          "application/vnd.emc.documentum+json"
        ]
      }
    },
    "about": {
      "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/product-
info.json",
      "hints": {
        "allow": [
          "GET"
        ],
        "representations": [
          "application/xml",
          "application/json",
          "application/vnd.emc.documentum+xml",
          "application/vnd.emc.documentum+json"
        ]
      }
    }
  }
}

```

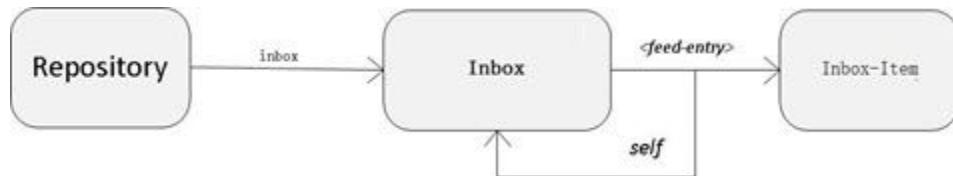


## 2.36 Inbox

The *Inbox* resource is used to get the notifications and workflow tasks of a user.

### 2.36.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

### 2.36.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the feed
- Feed Title: Inbox items
- Feed Updated: Server current time
- Supports POST: No
- Entry: “[Inbox Item](#)” on page 343 resource
- Entry ID: Entry URI
- Entry Title: Name of the inbox item
- Entry Updated: The date and time that the inbox item was sent to the user

### 2.36.3 Link relations

Link relation	Description	Resource reference
self	Link to this inbox	<a href="#">“Inbox” on page 337</a>

### 2.36.4 Operations

#### 2.36.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the inbox

## 2.36.4.2 Get Inbox

Retrieves the notifications and workflow tasks that are in the current user's inbox.

### 2.36.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, inline, page, items-per-page  item-type	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.  Specifies what kind of items the Inbox should return.	Integer	<p>It can be any of the following value bitwise OR'ed together.</p> <ul style="list-style-type: none"> <li>• 0xFFFF: (<i>DF_ALL</i>) including all notifications and tasks</li> <li>• 0x0001: (<i>DF_NOTIFICATION</i>) including only notifications but not queued notifications</li> <li>• 0x0002: (<i>DF_QUEUED</i>) including only queued notifications</li> <li>• 0x0004: (<i>DF_WORKFLOWTASK</i>) including only workflow tasks</li> <li>• 0x0008: (<i>DF_ROUTERTASK</i>) including only router task</li> </ul>	<p>The default is to return all notifications and workflow tasks.</p> <p>A default value of 7, which is the bitwise value of: 0x001   0x002   0x004</p>



**Note:** The *include-total*, *sort*, *view* and *filter* parameters are not supported.

#### **2.36.4.2.2 Request media types**

N/A

#### **2.36.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### **2.36.4.2.4 Request body**

N/A

#### **2.36.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.36.4.2.6 Response media types**

- application/xml
- application/atom+xml
- application/json
- application/vnd.emc.documentum+json
- application/hal+json

#### **2.36.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.36.4.2.8 Response body

➡ Example 2-129: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/inbox</id>
  <title>Inbox items</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2019-02-27T07:34:37.666+00:00</updated>
  <link href="http://localhost:8080/dctm-rest/repositories/REPO/inbox"
        rel="self"/>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/inbox/
      1b000001800011fb</id>
    <title>Activity-2</title>
    <summary>ACQUIRED</summary>
    <updated>2018-12-25T09:18:42.000+00:00</updated>
    <published>2018-12-25T09:18:42.000+00:00</published>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/inbox/
        1b000001800011fb" rel="edit"/>
    <content
      src="http://localhost:8080/dctm-rest/repositories/REPO/inbox/
        1b000001800011fb" type="application/vnd.emc.documentum+xml"/>
  </entry>
</feed>
```



➡ Example 2-130: JSON response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/inbox",
  "title": "Inbox items",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-27T07:34:39.138+00:00",
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
        1b000001800011fb",
      "title": "Activity-2",
      "summary": "ACQUIRED",
      "updated": "2018-12-25T09:18:42.000+00:00",
      "published": "2018-12-25T09:18:42.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            inbox/1b000001800011fb"
        }
      ]
    }
  ]
}
```

```
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
1b000001800011fb"
        }
    }
}
```



#### Example 2-131: HAL response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/inbox",
    "title": "Inbox items",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-27T07:34:43.310+00:00",
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
1b000001800011fb",
                "title": "Activity-2",
                "summary": "ACQUIRED",
                "updated": "2018-12-25T09:18:42.000+00:00",
                "published": "2018-12-25T09:18:42.000+00:00",
                "_links": {
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
inbox/1b000001800011fb"
                    }
                }
            }
        ],
        "_links": {
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox"
            }
        }
    }
}
```

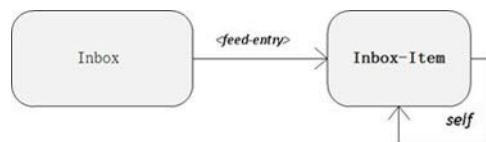


## 2.37 Inbox Item

The *Inbox Item* resource represents a notification or task in a user's notifications and workflow tasks.

### 2.37.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Inbox” on page 337 and About the Diagram on page 13.

### 2.37.2 Feed

No feed support.

### 2.37.3 Link relations

Link relation	Description	Resource reference
self	Link to itself	“Inbox Item” on page 343
workflow [1]	workflow	“Workflow” on page 906
workflow-activity [1]	workflow activity	“Workflow Activity” on page 935
task-packages [1]	task packages	“Task Packages” on page 719
delete [1]	delete notification item	“Inbox Item” on page 343
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.37.4 Operations

### 2.37.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the properties of the inbox item
PUT	Performs actions on the workflow task item
DELETE	Removes a notification from the inbox

### 2.37.4.2 Get an Inbox Item

Retrieves information about a single inbox item.

#### 2.37.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.37.4.2.2 Request media types

N/A

#### 2.37.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.37.4.2.4 Request body

N/A

#### 2.37.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.37.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.37.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.37.4.2.8 Response body

##### ➡ Example 2-132: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow-task xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <process-id>4b0000018000236e</process-id>
    <is-signoff-required>false</is-signoff-required>
    <is-repeatable>true</is-repeatable>
    <is-delegable>true</is-delegable>
    <is-end-activity>false</is-end-activity>
    <next-forward-port-names>
        <item>Output:2</item>
        <item>Output:0</item>
    </next-forward-port-names>
    <next-reject-port-names>
        <item>Output:1</item>
    </next-reject-port-names>
    <is-distribution-listtsak>false</is-distribution-listtsak>
    <is-group-owned>false</is-group-owned>
    <runtime-state>ACQUIRED</runtime-state>
    <is-failed-autotask>false</is-failed-autotask>
    <is-manual-transition>false</is-manual-transition>
    <instruction/>
    <is-sequential-a-activity>false</is-sequential-a-activity>
    <has-self-reject-link>false</has-self-reject-link>
    <is-first-performer>false</is-first-performer>
    <is-last-performer>false</is-last-performer>
    <has-forward-activities>true</has-forward-activities>
    <has-reject-activities>true</has-reject-activities>
    <queue-item-id>1b000001800011fb</queue-item-id>
    <activity-id>4c000001800025c8</activity-id>
    <activity-name>Activity-2</activity-name>
    <is-performer-assignment-required>false</is-performer-assignment-required>
    <complete-error-status>COMPLETE_UNKNOWN</complete-error-status>
    <task-name>Activity-2</task-name>
    <task-type>WORKFLOWTASK</task-type>
    <task-state>ACQUIRED</task-state>
    <priority>0</priority>
```

```

<sent-by>dadmin</sent-by>
<sent-date>2018-12-25T09:18:42.000+00:00</sent-date>
<item-id>4a00000180000102</item-id>
<workflow-id>4d00000180000100</workflow-id>
<workflow-task-type>NORMAL</workflow-task-type>
<performer>dadmin</performer>
<stamp>1b000001800011fb</stamp>
<object-id>1b000001800011fb</object-id>
<links>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/inbox/
              1b000001800011fb"
        rel="self"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
              4d00000180000100"
        rel="http://identifiers.emc.com/linkrel/workflow"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-activities/4c000001800025c8"
        rel="http://identifiers.emc.com/linkrel/workflow-activity"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/work-items/
              4a00000180000102/packages"
        rel="http://identifiers.emc.com/linkrel/task-packages"/>
</links>
</workflow-task>

```



### Example 2-133: JSON response

```
{
    "name": "workflow-task",
    "process-id": "4b0000018000236e",
    "is-signoff-required": false,
    "is-repeatable": true,
    "is-delegable": true,
    "is-end-activity": false,
    "next-forward-activities": [
        "Activity-4",
        "Activity-3"
    ],
    "next-forward-port-names": [
        "Output:2",
        "Output:0"
    ],
    "next-reject-port-names": [
        "Output:1"
    ],
    "is-distribution-listtsak": false,
    "is-group-owned": false,
    "runtime-state": "ACQUIRED",
    "is-failed-autotask": false,
    "is-manual-transition": false,
    "instruction": "",
    "is-sequential-a-activity": false,
    "has-self-reject-link": false,
    "is-first-performer": false,
    "is-last-performer": false,
}
```

```

    "has-forward-activities": true,
    "has-reject-activities": true,
    "queue-item-id": "1b000001800011fb",
    "activity-id": "4c000001800025c8",
    "activity-name": "Activity-2",
    "is-performer-assignment-required": false,
    "complete-error-status": "COMPLETE_UNKNOWN",
    "task-name": "Activity-2",
    "task-type": "WORKFLOWTASK",
    "task-state": "ACQUIRED",
    "priority": 0,
    "sent-by": "dmadmin",
    "sent-date": "2018-12-25T09:18:42.000+00:00",
    "item-id": "4a00000180000102",
    "workflow-id": "4d00000180000100",
    "workflow-task-type": "NORMAL",
    "performer": "dmadmin",
    "stamp": "1b000001800011fb",
    "object-id": "1b000001800011fb",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
                    1b000001800011fb"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/workflow",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflows/4d00000180000100"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c8"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/task-packages",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
                    4a00000180000102/packages"
        }
    ]
}

```



#### Example 2-134: HAL response

```
{
    "name": "workflow-task",
    "process-id": "4b0000018000236e",
    "is-signoff-required": false,
    "is-repeatable": true,
    "is-delegable": true,
    "is-end-activity": false,
    "next-forward-port-names": [
        "Output:2",
        "Output:0"
    ],
    "next-reject-activities": [
        "Activity-1"
    ],
    "next-reject-port-names": [
        "Output:1"
    ],
    "is-distribution-listtsak": false,
}
```

```
        "is-group-owned": false,
        "runtime-state": "ACQUIRED",
        "is-failed-autotask": false,
        "is-manual-transition": false,
        "instruction": "",
        "is-sequential-a-activity": false,
        "has-self-reject-link": false,
        "is-first-performer": false,
        "is-last-performer": false,
        "has-forward-activities": true,
        "has-reject-activities": true,
        "queue-item-id": "1b000001800011fb",
        "activity-id": "4c000001800025c8",
        "activity-name": "Activity-2",
        "is-performer-assignment-required": false,
        "complete-error-status": "COMPLETE_UNKNOWN",
        "task-name": "Activity-2",
        "task-type": "WORKFLOWTASK",
        "task-state": "ACQUIRED",
        "priority": 0,
        "sent-by": "dmadmin",
        "sent-date": "2018-12-25T09:18:42.000+00:00",
        "item-id": "4a0000018000102",
        "workflow-id": "4d00000180000100",
        "workflow-task-type": "NORMAL",
        "performer": "dmadmin",
        "stamp": "1b000001800011fb",
        "object-id": "1b000001800011fb",
        "_links": {
            "http://identifiers.emc.com/linkrel/workflow": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
4d00000180000100"
            },
            "http://identifiers.emc.com/linkrel/workflow-activity": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
workflow-activities/4c000001800025c8"
            },
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
1b000001800011fb"
            },
            "http://identifiers.emc.com/linkrel/task-packages": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
4a0000018000102/packages"
            }
        }
    }
```



### 2.37.4.3 Performing actions on a workflow task item

Some actions can be performed on a workflow task item.

HTTP Method	Query Parameters	Description	Data Type	Value Range	Default Value
PUT	action	<p>What action will be done on the workflow task inbox item. Currently, the following actions are supported:</p> <ul style="list-style-type: none"> <li>• ACQUIRE: Acquire this workflow task</li> <li>• FORWARD: Set next forward activity for this task, and complete current task, activity name is required</li> <li>• REJECT: Set next reject activity for this task, and complete current task, activity name is required</li> <li>• DELEGATE: Delegate this task to another user, and complete current task, delegate-user is required</li> <li>• REPEAT: Set the repeaters for the task, and complete current task, repeat-users is required</li> <li>• SIGNOFF: Signoff and complete the task, signoff-password is required</li> <li>• ASSIGN: Set performers for activity, activity and assign-performers is required</li> <li>• COMPLETE: Complete the workflow task</li> </ul> <p>The parameter is required</p>	string	one supported action	null
	delegate-user	<p>URL encoded user name, with charset UTF-8, of the user to be delegated</p> <p>Mandatory when action=DELEGATE. For other actions, the value is ignored</p>	string		null

HTTP Method	Query Parameters	Description	Data Type	Value Range	Default Value
	repeat-users	URL encoded user names and/or group names to repeat the task  More values are separated by the comma, e.g. Usera, userb, userc  Mandatory when action=REPEAT. For other actions, the value is ignored	string		null
	assign-performers	URL encoded user names and/or group names to be assigned to the target activity.  More values are separated by the comma, e.g. Usera, userb, userc  Mandatory when action=ASSIGN. For other actions, the value is ignored	string		null
	signoff-password	The signoff password  Mandatory when action=SIGNOFF. For other actions, the value is ignored	string		null
	activity	The next forward or reject activity name.  Optional when action=FORWARD or REJECT. For other actions, the value is ignored	string		null

#### 2.37.4.3.1 Request media types

N/A

#### 2.37.4.3.2 Request headers

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.37.4.3.3 Request body

N/A

#### 2.37.4.3.4 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.37.4.3.5 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.37.4.3.6 Response status

- 200 - Action performed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.37.4.3.7 Response body

➡ **Example 2-135: XML response**

The same response as in the GET operation.



➡ **Example 2-136: JSON response**

The same response as in the GET operation.



➡ **Example 2-137: HAL response**

The same response as in the GET operation.



#### 2.37.4.4 Delete a notification item

Deletes a specified notification item from the inbox.

##### 2.37.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.37.4.4.2 Request media types

N/A

##### 2.37.4.4.3 Request headers

- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.37.4.4.4 Request body

N/A

##### 2.37.4.4.5 Response headers

- N/A

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.37.4.4.6 Response media types

N/A

##### 2.37.4.4.7 Response status

- 204 - Successfully deleted
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.37.4.4.8 Response body

N/A

## 2.38 Job(s)

### 2.38.1 Job

The *Job* resource represents the metadata of a single job.

#### 2.38.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Job” on page 353 and [About the Diagram on page 13](#).

#### 2.38.1.2 Feed

No feed support.

#### 2.38.1.3 Link relations

Link relation	Description	Resource reference
self	Link to itself	<a href="#">“Job” on page 353</a>
edit	Link used to edit the metadata of a job and enable scheduling of a job	<a href="#">“Job” on page 353</a>
delete [1]	Link used to delete the Job instance	<a href="#">“Job” on page 353</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

#### 2.38.1.4 Operations

#### 2.38.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieve a job
POST	Update a job
DELETE	Delete a job

#### 2.38.1.4.2 Get Job metadata

Retrieves the metadata of a single specified job.

##### 2.38.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.38.1.4.2.2 Request media types

N/A

##### 2.38.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.38.1.4.2.4 Request body

N/A

##### 2.38.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.38.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.38.1.4.2.8 Response body

##### ➡ Example 2-138: Request

```
GET http://localhost:8080/dctm-rest/repositories/documentum1/jobs/
    080004d28000035b
Accept:application/xml
Authorization:Basic ZG1hZG1pbjpwYXNzd29yZA==
```



##### ➡ Example 2-139: Response

```
<dm:job xsi:type="dm_job"
    definition="http://localhost:8080/dctm-rest/repositories/documentum1/
        types/dm_job">
<dm:properties>
    <dm:r_object_id>080004d28000035b</dm:r_object_id>
    <dm:object_name>dm_DMClean</dm:object_name>
    <dm:r_object_type>dm_job</dm:r_object_type>
    <dm:subject>Executes dmclean on a schedule</dm:subject>
    <dm:title>Content</dm:title>
    <dm:owner_name>Administrator</dm:owner_name>
    <dm:method_trace_level></dm:method_trace_level>
    <dm:target_server>documentum1.documentum1@cs1671</dm:target_server>
    <dm:is_inactive>true</dm:is_inactive>
    <dm:start_date>2019-11-18T17:30:00.000+00:00</dm:start_date>
    <dm:inactivate_after_failure>false</dm:inactivate_after_failure>
    <dm:a_next_invocation>2019-11-18T17:30:00.000+00:00</
dm:a_next_invocation>
    <dm:run_mode>4</dm:run_mode>
    <dm:max_iterations>0</dm:max_iterations>
    <dm:expiration_date>2027-11-18T17:30:00.000+00:00</dm:expiration_date>
    <dm:method_name>dm_DMClean</dm:method_name>
    <dm:method_arguments>
        <dm:item>-window_interval 120</dm:item>
        <dm:item>-queueperson</dm:item>
        <dm:item>-clean_content TRUE</dm:item>
        <dm:item>-clean_note TRUE</dm:item>
        <dm:item>-clean_acl TRUE</dm:item>
        <dm:item>-clean_wf_template TRUE</dm:item>
        <dm:item>-clean_now TRUE</dm:item>
        <dm:item>-clean_castore FALSE</dm:item>
        <dm:item>-clean_aborted_wf FALSE</dm:item>
        <dm:item>-clean_deleted_lwso TRUE</dm:item>
```

```

<dm:item>-clean_content_in_parallel FALSE</dm:item>
<dm:item>-parallel_degree 5</dm:item>
<dm:item>-clean_wf_method_exec_result FALSE</dm:item>
<dm:item>-batch_size 2000</dm:item>
<dm:item>-query_batch_size 0</dm:item>
</dm:method_arguments>
</dm:properties>
<links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/documentum1/
jobs/
            080004d28000035b" />
    <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/documentum1/
jobs/
            080004d28000035b" />
    <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/repositories/
documentum1/
            jobs/080004d28000035b" />
</links>
</dm:job>

```



#### 2.38.1.4.3 Update a Job

Updates the metadata of a Job by providing attributes.

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.38.1.4.3.1 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

##### 2.38.1.4.3.2 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.1.4.3.3 Request body

➡ Example 2-140: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:dm_job xmlns:dm="http://ns.emc.com/documentum"
             xmlns="http://www.w3.org/2005/Atom">
    <dm:properties>
        <dm:object_name>dm_DMClean</dm:object_name>
        <dm:start_date>2019-11-20T05:46:08.000+00:00</dm:start_date>
        <dm:max_iterations>1</dm:max_iterations>
        <dm:method_name>my_method</dm:method_name>
        ...
    </dm:properties>
</dm:dm_job>
```



#### 2.38.1.4.3.4 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.1.4.3.5 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.38.1.4.3.6 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.38.1.4.3.7 Response body

➡ Example 2-141: XML Response

```
<dm:job xsi:type="dm_job"
         definition="http://localhost:8080/dctm-rest/repositories/documentum1/
                     types/dm_job">
    <dm:properties>
        <dm:r_object_id>080004d28000035b</dm:r_object_id>
        <dm:object_name>dm_DMClean</dm:object_name>
        <dm:r_object_type>dm_job</dm:r_object_type>
        <dm:subject>Executes dmclean on a schedule</dm:subject>
        <dm:title>Content</dm:title>
        <dm:owner_name>Administrator</dm:owner_name>
        <dm:method_trace_level>0</dm:method_trace_level>
        <dm:target_server>documentum1.documentum1@cs1671</dm:target_server>
        <dm:is_inactive>true</dm:is_inactive>
        <dm:start_date>2019-11-20T05:46:08.000+00:00</dm:start_date>
```

```
<dm:inactivate_after_failure>false</dm:inactivate_after_failure>
<dm:a_next_invocation>2019-11-18T17:30:00.000+00:00</
dm:a_next_invocation>
<dm:run_mode>4</dm:run_mode>
<dm:max_iterations>0</dm:max_iterations>
<dm:expiration_date>2027-11-18T17:30:00.000+00:00</dm:expiration_date>
<dm:method_name>my_method</dm:method_name>
<dm:method_arguments>
    <dm:item>-window_interval 120</dm:item>
    <dm:item>-queueperson</dm:item>
    <dm:item>-clean_content TRUE</dm:item>
    <dm:item>-clean_note TRUE</dm:item>
    <dm:item>-clean_acl TRUE</dm:item>
    <dm:item>-clean_wf_template TRUE</dm:item>
    <dm:item>-clean_now TRUE</dm:item>
    <dm:item>-clean_castore FALSE</dm:item>
    <dm:item>-clean_aborted_wf FALSE</dm:item>
    <dm:item>-clean_deleted_lwso TRUE</dm:item>
    <dm:item>-clean_content_in_parallel FALSE</dm:item>
    <dm:item>-parallel_degree 5</dm:item>
    <dm:item>-clean_wf_method_exec_result FALSE</dm:item>
    <dm:item>-batch_size 2000</dm:item>
    <dm:item>-query_batch_size 0</dm:item>
</dm:method_arguments>
</dm:properties>
<links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/documentum1/
jobs/
          080004d28000035b" />
    <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/documentum1/
jobs/
          080004d28000035b" />
    <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/repositories/documentum1/
jobs/
          080004d28000035b" />
</links>
</dm:job>
```



#### 2.38.1.4.4 Delete a Job

Deletes a specified Job.

#### 2.38.1.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.38.1.4.4.2 Request media types

N/A

#### 2.38.1.4.4.3 Request headers

- N/A

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.1.4.4.4 Request body

N/A

#### 2.38.1.4.4.5 Response headers

- N/A

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.1.4.4.6 Response media types

N/A

#### 2.38.1.4.4.7 Response status

- 204 - Successfully deleted
- 400, 401, 403, 404, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.38.1.4.4.8 Response body

N/A

### 2.38.2 Jobs

The `Jobs` resource represents a collection of `Job` instances in a repository.

#### 2.38.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

#### 2.38.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of jobs
- Feed Updated: The current time
- Entry: “[Job](#)” on page 353
- Supports POST: Yes

#### 2.38.2.3 Link relations

Link relation	Description	Resource reference
self	The URI of the collection of Jobs.	<a href="#">“Jobs” on page 360</a>
first, last, next, previous	Pagination links.	<a href="#">“Jobs” on page 360</a>

## 2.38.2.4 Operations

### 2.38.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of Job instances.
POST	Used to create new Jobs instances.

### 2.38.2.4.2 Get a Jobs collection

Retrieve a collection of Jobs along with their filters and pagination.

#### 2.38.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value Range	Default value
GET	inline	Determines whether to show content (the object instance) in the atom entry for a collection.	boolean	true - returns an object instance and embeds the object instance into the entry's content.  false - does not return an object instance.	false
	sort	Only groups that contain a sub group (a dm_group instance) whose group_name property equals to the specified string are returned in the result.	string	string	null

HTTP method	Query parameters	Description	Data type	Value Range	Default value
	page	Determines the number of the page to be served; when the result feed is paged, an element <dm:page>x</dm:page> is displayed under the feed root.	integer	>=1	1 for example, items_per_page=200&page=2 returns items 201 to 400
	items-per-page	Determines the number of entries per page; when the result feed is paged, an element <dm:items-per-page>xx</dm:items-per-page> is displayed under the feed root.	integer	>=1	100
	include-total	Indicates whether to calculate the total count of feed items that are returned on one page.  When true, a <dm:total>xx</dm:total> element displays under the feed root.	boolean	true - the total count of feed items is calculated by server and returned.  false - not calculate & return the total count.	false

HTTP method	Query parameters	Description	Data type	Value Range	Default value
	view	Defines the properties to return and that are displayed.	string	The pattern is: ? view=(:view-name)?(, column)* - (:view-name) and (columns)* must be mutually displayed.  exclusive - names of predefined views start with ":"; the following view-name values are defined: - :all - :default - if no view-name is specified, names of properties or predefined views should be returned, separated by comma (',').	:default

HTTP method	Query parameters	Description	Data type	Value Range	Default value
	links	Determines whether link relations are returned for this object instance.	boolean	true or false. <ul style="list-style-type: none"><li>• links = true is default</li><li>• links = false means "do not send me any links that are expensive to calculate, but it is fine if you send me ones that are cheap to calculate."</li></ul>	true

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.38.2.4.2.2 Using filters

- The following filter queries can be used to get Jobs by name:

```
/repositories/<repositoryName>/jobs?filter=object_name='dm_AsyncronousWrite'  
/repositories/<repositoryName>/jobs?filter=starts-with(object_name,"dm_Asy")
```

- The following filter query can be used to get Jobs by title:

```
/repositories/<repositoryName>/jobs?filter=title='CA Store'
```

List of the available Job types:

- CA Store (EMC Centera and NetApp SnapLock stores)
- Content
- Data Dictionary
- Distributed Content
- Docbase
- Federation
- Fulltext
- Other
- Replication

- Workflow

#### 2.38.2.4.2.3 Request media types

N/A

#### 2.38.2.4.2.4 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.2.4.2.5 Request body

N/A

#### 2.38.2.4.2.6 Response headers

- Content-Type
- Content-Length

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.2.4.2.7 Response media types

N/A

#### 2.38.2.4.2.8 Response status

- 200 - Request successful
- 400, 401, 403, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.38.2.4.2.9 Response body

- The body contains a list of Jobs that are in the repository.
- Each Job metadata contains the following attributes:
  1. **Id:** The Id of the Job.
  2. **Title:** The name of the Job.
  3. **Summary:** A brief description of the Job.

 **Example 2-142: Request**

```
GET http://localhost:9089/dctm-rest/repositories/documentum1/jobs?inline=true
Accept:application/atom+xml
Authorization:Basic ZG1hZG1pbjpwYXNzd29yZA==
```



 **Example 2-143: XML Response**

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:9089/dctm-rest/repositories/documentum1/jobs</id>
    <title>jobs</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2020-01-03T05:50:06.140+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        100
    </dm:items-per-page>
    <link rel="self"
          href="http://localhost:9089/dctm-rest/repositories/documentum1/jobs" />
    <link rel="first"
          href="http://localhost:9089/dctm-rest/repositories/documentum1/
                jobs?items-per-page=100&page=1" />
    <entry>
        <id>http://localhost:9089/dctm-rest/repositories/documentum1/jobs/
            080004d28000035f</id>
        <title>dm_AuditMgt</title>
        <summary>Removes old audit trail entries</summary>
        <updated>2019-11-20T05:46:08.000+00:00</updated>
        <published>2019-11-20T05:46:08.000+00:00</published>
        <content xmlns:dm="http://identifiers.emc.com/vocab/documentum">
            <dm:job xsi:type="dm_job"
                    definition="http://localhost:9089/dctm-rest/repositories/
                        documentum1/types/dm_job">
                <dm:properties>
                    <dm:r_object_id>080004d28000035f</dm:r_object_id>
                    <dm:r_object_type>dm_job</dm:r_object_type>
                    <dm:object_name>dm_AuditMgt</dm:object_name>
                    <dm:subject>Removes old audit trail entries</dm:subject>
                    <dm:title>Docbase</dm:title>
                </dm:properties>
                <dm:links>
                    <dm:link rel="self"
                            href="http://localhost:9089/dctm-rest/repositories/
                                documentum1/jobs/080004d28000035f" />
                    <dm:link rel="edit"
                            href="http://localhost:9089/dctm-rest/repositories/
                                documentum1/jobs/080004d28000035f" />
                </dm:links>
            </dm:job>
        </content>
    </entry>
    <entry>
        <id>http://localhost:9089/dctm-rest/repositories/documentum1/jobs/
            080004d28000036f</id>
        <title>dm_ContentReplication</title>
        <summary>
            Replicates content from specified source (or all) to target site
        </summary>
    </entry>
</feed>
```

```

</summary>
<updated>2020-01-03T09:29:17.060+00:00</updated>
<published>2020-01-03T09:29:17.060+00:00</published>
<content xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    <dm:job xsi:type="dm_job"
        definition="http://localhost:8080/dctm-rest/repositories/
            documentum1/types/dm_job">
        <dm:properties>
            <dm:r_object_id>080004d28000036f</dm:r_object_id>
            <dm:r_object_type>dm_job</dm:r_object_type>
            <dm:object_name>dm_ContentReplication</dm:object_name>
            <dm:subject>Replicates content from specified source (or all) to
                target site</dm:subject>
            <dm:title>Replication</dm:title>
        </dm:properties>
        <dm:links>
            <dm:link rel="self"
                href="http://localhost:9089/dctm-rest/repositories/
                    documentum1/jobs/080004d28000036f"/>
            <dm:link rel="edit"
                href="http://localhost:9089/dctm-rest/repositories/
                    documentum1/jobs/080004d28000036f"/>
        </dm:links>
    </dm:job>
</content>
</entry>
</feed>

```



#### Example 2-144: Response for starts-with filter

```

GET http://localhost:9089/dctm-rest/repositories/documentum1/
    jobs?filter=starts-with(object_name, "dm_Au")
Accept:application/atom+xml
Authorization:Basic ZG1hZG1pbjpwYXNzd29yZA==

```



#### Example 2-145: XML Response for starts-with filter

```

<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:9089/dctm-rest/repositories/documentum1/jobs</id>
    <title>jobs</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2020-01-03T05:50:06.140+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        100
    </dm:items-per-page>
    <link rel="self"
          href="http://localhost:9089/dctm-rest/repositories/documentum1/jobs"/>
    <entry>
        <id>http://localhost:9089/dctm-rest/repositories/documentum1/jobs/
            080004d28000035f</id>
        <title>dm_AuditMgt</title>
        <summary>Removes old audit trail entries</summary>
        <updated>2019-11-20T05:46:08.000+00:00</updated>
    </entry>

```

```
<published>2019-11-20T05:46:08.000+00:00</published>
<content xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    <dm:job xsi:type="dm_job"
        definition="http://localhost:9089/dctm-rest/repositories/
            documentum1/types/dm_job">
        <dm:properties>
            <dm:r_object_id>080004d28000035f</dm:r_object_id>
            <dm:r_object_type>dm_job</dm:r_object_type>
            <dm:object_name>dm_AuditMgt</dm:object_name>
            <dm:subject>Removes old audit trail entries</dm:subject>
            <dm:title>Docbase</dm:title>
        </dm:properties>
        <dm:links>
            <dm:link rel="self"
                href="http://localhost:9089/dctm-rest/repositories/
                    documentum1/jobs/080004d28000035f"/>
            <dm:link rel="edit"
                href="http://localhost:9089/dctm-rest/repositories/
                    documentum1/jobs/080004d28000035f"/>
        </dm:links>
    </dm:job>
</content>
</entry>
</feed>
```



#### 2.38.2.4.3 Create a Job

Creates a Job in the repository.

##### 2.38.2.4.3.1 HTTP Request method and query parameters

POST

##### 2.38.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

##### 2.38.2.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.2.4.3.4 Request body

The following parameters are mandatory to create a Job:

- *method\_name*: Values are any valid string.
- *start\_date*: Values are any valid date *year-month-day*
- *run\_mode*: Values are integers that range from 1 - 9

The following illustrates how to use *run\_mode* values:

- 1: Every *x* minutes
- 2: Every *x* hours
- 3: Every *x* days
- 4: Every *x* weeks
- 5: Every *x* months
- 7: Every *x* day of week
- 8: Every *x* day of month
- 9: Every *x* day of year

Where *x* represents the value set for the *run\_interval* property.

- *run\_interval*: Values must be greater than 1
- *max\_iterations* or *expiration\_date*

#### Example 2-146: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:dm_job xmlns:dm="http://ns.emc.com/documentum"
             xmlns="http://www.w3.org/2005/Atom">
    <dm:properties>
        <dm:object_name>acme</dm:object_name>
        <dm:start_date>acme</dm:start_date>
        <dm:max_iterations>1</dm:max_iterations>
        <dm:method_name>my_method</dm:method_name>
        .....
    </dm:properties>
</dm:dm_job>
```



#### 2.38.2.4.3.5 Response headers

- Location

The URL of the Job created.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.38.2.4.3.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json

#### 2.38.2.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.38.2.4.3.8 Response body

N/A

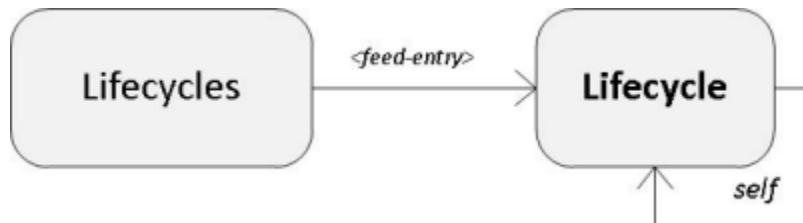
### 2.39 Lifecycle (s)

#### 2.39.1 Lifecycle

This resource represents a single Lifecycle object.

##### 2.39.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* ["Lifecycles" on page 385](#) and [About the Diagrams on page 13](#).

### 2.39.1.2 Feed

No feed support.

### 2.39.1.3 Link relations

Link relation	Description	Resource reference
self	This Lifecycle resource	<a href="#">"Lifecycle" on page 370</a>
permission-set	The permission set of the lifecycle	<a href="#">"Permission Set" on page 507</a>
permissions	The lifecycle permissions	<a href="#">"Permissions" on page 503</a>
type	The type of the type inclusion	<a href="#">"Type" on page 727</a>
owner	The owner user resource	<a href="#">"User" on page 766</a>

### 2.39.1.4 Operations

#### 2.39.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a single Lifecycle object.

#### 2.39.1.4.2 Get a Lifecycle

Retrieve a single lifecycle object.

##### 2.39.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.39.1.4.2.2 Request media types

N/A

##### 2.39.1.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.39.1.4.2.4 Request body**

N/A

**2.39.1.4.2.5 Response headers**

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.39.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.39.1.4.2.7 Response status**

- 200 – Retrieved successfully.
- 400, 401, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.39.1.4.2.8 Response body****Example 2-147: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<lifecycle xmlns="http://identifiers.emc.com/vocab/documentum"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>4600000580004abc</id>
    <name>my_lifecycle</name>
    <title>xxx</title>
    <subject>xxx</subject>
    <keywords>
        <keyword>xxx</keyword>
    </keywords>
    <implementation-type>JAVA</implementation-type>
    <owner>dadmin</owner>
    <created>2017-03-30T07:18:36.000+00:00</created>
    <modified>2017-03-30T07:18:36.000+00:00</modified>
    <version-labels>
        <version-label>1.0</version-label>
        <version-label>CURRENT</version-label>
    </version-labels>
    <alias-sets>
        <alias-set>660000058000010a</alias-set>
    </alias-sets>
    <type-inclusions>
        <type-inclusion>
            <type>dm_sysobject</type>
            <include-subtypes>false</include-subtypes>
        </type-inclusion>
    </type-inclusions>

```

```

        </type-inclusion>
        <type-inclusion>
            <type>dm_document</type>
            <include-subtypes>true</include-subtypes>
        </type-inclusion>
    </type-inclusions>
    <user-validation-service>
        <id>0b00000580004df1</id>
        <name>lifecyclemodule</name>
        <primary-class>
            com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor
        </primary-class>
    </user-validation-service>
    <app-validation>
        <id>0b00000580004df1</id>
        <name>lifecyclemodule</name>
        <version></version>
    </app-validation>
    <states>
        <state>
            <name>draft</name>
            <type></type>
            <description>draft version</description>
            <exceptional>false</exceptional>
            <exception-state></exception-state>
            <allow-attach>true</allow-attach>
            <allow-schedule>true</allow-schedule>
            <allow-return-to-base>false</allow-return-to-base>
            <allow-demote>false</allow-demote>
            <index>0</index>
            <no>0</no>
            <return-conditions>
                <return-condition>ALL</return-condition>
            </return-conditions>
            <type-override-id></type-override-id>
            <entry-criteria>
                <id>550000058000093b</id>
                <expression>((subject = "ok"))</expression>
            </entry-criteria>
            <action>
                <id>0800000580004e05</id>
                <name>
                    review_fmuzh1p0kpq8ilz5ifvexz actionset procedure for policy
a_docbasic_lifecycle
                </name>
            </action>
            <user-criteria>
                <id></id>
                <name></name>
                <version></version>
            </user-criteria>
            <user-action>
                <id></id>
                <name></name>
                <version></version>
            </user-action>
            <user-post-action>
                <id></id>
                <name></name>
                <version></version>
            </user-post-action>
            <user-criteria-service>

```

```

<id>0b00000580004df1</id>
<name>lifecyclemodule</name>
<primary-class>
    com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor
</primary-class>
</user-criteria-service>
<user-action-service>
    <id>0b00000580004df1</id>
    <name>lifecyclemodule</name>
    <primary-class>
        com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor
    </primary-class>
</user-action-service>
<user-post-service>
    <id>0b00000580004df1</id>
    <name>lifecyclemodule</name>
    <primary-class>
        com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor
    </primary-class>
</user-post-service>
<system-action>
    <id></id>
    <name></name>
    <primary-class></primary-class>
</system-action>
</state>
</states>
<status>INSTALLED</status>
<links>...</links>
</lifecycle>

```



#### Example 2-148: JSON response

```
{
    "id" : "4600000580004abc",
    "name" : "my_lifecycle",
    "title" : "xxx",
    "subject" : "xxx",
    "keywords" : "xxx",
    "implementation-type" : "JAVA",
    "owner" : "dmadmin",
    "created" : "2017-03-30T07:18:36.000+00:00",
    "modified" : "2017-03-30T07:18:36.000+00:00",
    "version-labels" : [ "1.0", "CURRENT" ],
    "alias-set" : [ "660000058000010a" ],
    "type-inclusions" : [ {
        "type" : "dm_sysobject",
        "include-subtypes" : false
    }, {
        "type" : "dm_document",
        "include-subtypes" : true
    } ],
    "user-validation-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-class" :
            "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    },
    "app-validation" : {

```

```

        "id" : "0800000580004e05",
        "name" : ""
    },
    "states" : [
        {
            "name" : "draft",
            "type" : "",
            "description" : "draft version",
            "exceptional" : false,
            "exception-state" : "",
            "allow-attach" : true,
            "allow-schedule" : true,
            "allow-return-to-base" : false,
            "allow-demote" : false,
            "index" : 0,
            "return-conditions" : [ "ALL" ],
            "entry-criteria" : {
                "id" : "550000058000093b",
                "expression" : "((subject = \"ok\") Or (subject = \"action fail\")) Or (subject = \"post fail\"))"
            },
            "action" : {
                "id" : "0800000580004e05",
                "name" : "review_fmuzh1p0kpq8ilz5ifvexz actionset procedure for policy
                    a_docbasic_lifecycle",
                "versin" : ""
            },
            "user-criteria" : {
                "id" : "",
                "name" : "",
                "versin" : ""
            },
            "user-action" : {
                "id" : "",
                "name" : "",
                "versin" : ""
            },
            "user-post-action" : {
                "id" : "",
                "name" : "",
                "versin" : ""
            },
            "user-criteria-service" : {
                "id" : "0b00000580004df1",
                "name" : "lifecyclemodule",
                "primary-class" :
"com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
            },
            "user-action-service" : {
                "id" : "0b00000580004df1",
                "name" : "lifecyclemodule",
                "primary-class" :
"com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
            },
            "user-post-service" : {
                "id" : "0b00000580004df1",
                "name" : "lifecyclemodule",
                "primary-class" :
"com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
            },
            "system-action" : {
                "id" : "",
                "name" : ""
            }
        }
    ]
}

```

```

        "primary-class" : ""
    },
],
"type-override-id" : "",
"status" : "INSTALLED",
"links" : [xxx]
}

```



#### 2.39.1.4.2.9 Lifecycle Property Mappings

The following table shows the mapping between OpenText Documentum CM properties and Lifecycle properties.

**Table 2-1: Property mappings: OpenText Documentum CM to Lifecycle**

Lifecycle property name				Lifecycle property type	DCTM property name	DCTM property type
id				string	r_object_id	dm_policy
name				string	object_name	dm_policy
title				string	title	dm_policy
subject				string	subject	dm_policy
keywords	keyword			string	keywords	dm_policy
implementation-type				string	java_methods	dm_policy
owner				string	owner_name	dm_policy
created				string	r_creation_date	dm_policy
modified				string	r_modify_date	dm_policy
version-labels	version-label			string	r_version_label	dm_policy
alias-sets	alias-set			string	alias_set_ids	dm_policy
type-inclusions	type-inclusion	type		string	included_type	dm_policy
		include-subtype		string	include_subtypes	dm_policy
user-validation-service	id			string	r_object_id	dm_policy

Lifecycle property name				Lifecycle property type	DCTM property name	DCTM property type
	name			string	user_validation_service	dmc_module
	primary-class			string	primary_class	dm_policy
app-validation	id			string	app_validation_id	dm_procedure
	name			string	object_name	dm_policy
	version			string	app_validation_ver	dm_policy
states	state	name		string	state_name	dm_policy
		type		string	state_type	dm_policy
		description		string	state_description	dm_policy
		exceptional		string	state_class	dm_policy
		exception-state		string	exception_state	dm_policy
		allow-attach		string	allow_attach	dm_policy
		allow-schedule		string	allow_schedule	dm_policy
		allow-demote		string	allow_demote	dm_policy
		allow-return-to-base		string	return_to_base	dm_policy
		index		integer		
		no		integer	i_state_no	dm_policy
		return-conditions	return-condition	string	return_condition	dm_policy
		typeOverride-id		string	type_override_id	dm_policy
	entry-criteria	id		string	entry_criteria_id	dm_policy
		entry-criteria		string	expression_text	dm_func_expr

Lifecycle property name				Lifecycle property type	DCTM property name	DCTM property type
Lifecycle	Object	action	id	string	action_object_id	dm_policy
			name	string	object_name	dm_procedure
		user-criteria	id	string	user_criteria_id	dm_policy
			name	string	object_name	dm_procedure
			version	string	user_criteria_ver	dm_policy
		user-action	id	string	user_action_id	dm_policy
			name	string	object_name	dm_procedure
			version	string	user_action_ver	dm_policy
		user-post-action	id	string	user_postproc_id	dm_policy
			name	string	object_name	dm_procedure
			version	string	user_postproc_ver	dm_policy
			name	string	user_criteria_service	dm_policy
			primary-class	string	primary_class	dmc_module
Lifecycle	Object	user-action-service	id	string	r_object_id	dmc_module
			name	string	user_action_service	dm_policy
			primary-class	string	primary_class	dmc_module
		user-post-service	id	string	r_object_id	dmc_module
			name	string	user_postprocessing_service	dm_policy
			primary-class	string	primary_class	dmc_module

Lifecycle property name				Lifecycle property type	DCTM property name	DCTM property type
		system-action	id	string	r_object_id	dmc_modu le
			name	string	system_acti ons	dm_policy
			primary-class	string	primary_cl ass	dmc_modu le
status				string	r_definition _state	dm_policy

### 2.39.1.5 Additional examples for the response

A Lifecycle sample implemented in DocBasic.

#### ➡ Example 2-149: JSON response

```
{
  "id" : "4600000580004dfb",
  "name" : "a_docbasic_lifecycle",
  "owner" : "dmadmin",
  "created" : "2017-05-03T02:15:21.000+00:00",
  "modified" : "2017-05-03T02:15:25.000+00:00",
  "implementation-type" : "DOCBASIC",
  "version-labels" : [
    "1.0",
    "CURRENT"
  ],
  "status" : "INSTALLED",
  "type-inclusions" : [
    {
      "type" : "dm_document",
      "include-subtypes" : true
    }
  ],
  "states" : [
    {
      "name" : "draft",
      "index" : 0,
      "no" : 0,
      "exceptional" : false,
      "allow-attach" : true,
      "allow-schedule" : true,
      "allow-return-to-base" : false,
      "allow-demote" : false,
      "return-conditions" : [
        "ALL"
      ]
    },
    {
      "name" : "review",
      "index" : 1,
      "no" : 1,
      "exceptional" : false,
      "exception-state" : "fail",
      "allow-attach" : true,
      "allow-schedule" : true,
      "allow-return-to-base" : false,
      "allow-demote" : false,
      "return-conditions" : [
        "ALL"
      ]
    }
  ]
}
```

```
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : true,
    "allow-demote" : true,
    "return-conditions" : [
        "CHECKIN",
        "SAVE_AS_NEW",
        "BRANCH"
    ],
    "entry-criteria" : {
        "id" : "550000058000093b",
        "expression" : "(((subject = \"ok\") Or (subject = \"action fail\")) Or (subject = \"post fail\"))"
    },
    "action" : {
        "id" : "0800000580004e05",
        "name" : "review_fmuzh1p0kpq8ilz5ifvexz actionset procedure for policy
                    a_docbasic_lifecycle"
    }
},
{
    "name" : "publish",
    "index" : 2,
    "no" : 2,
    "exceptional" : false,
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : true,
    "allow-demote" : true,
    "return-conditions" : [
        "BRANCH"
    ],
    "entry-criteria" : {
        "id" : "550000058000093c",
        "expression" : "(((subject = \"ok\") Or (subject = \"action fail\")) Or (subject = \"post fail\"))"
    },
    "action" : {
        "id" : "0800000580004e06",
        "name" : "publish_2cp0x50b4s+anynusyzqm5 actionset procedure for
policy
                    a_docbasic_lifecycle"
    }
},
{
    "name" : "fail",
    "index" : 3,
    "no" : 3,
    "exceptional" : true,
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : false,
    "allow-demote" : false,
    "return-conditions" : [
        "ALL"
    ]
},
],
"links" : [
{
    "rel" : "self",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
```

```

        4600000580004dfb"
    },
    {
        "rel" : "http://identifiers.emc.com/linkrel/owner",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
    },
    {
        "rel" : "http://identifiers.emc.com/linkrel/type",
        "title" : "dm_document",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document"
    }
]
}

```



#### ➡ Example 2-150: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<lifecycle xmlns="http://identifiers.emc.com/vocab/documentum"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>4600000580004dfb</id>
    <name>a_docbasic_lifecycle</name>
    <owner>dmadmin</owner>
    <created>2017-05-03T02:15:21.000+00:00</created>
    <modified>2017-05-03T02:15:25.000+00:00</modified>
    <implementation-type>DOCBASIC</implementation-type>
    <version-labels>
        <version-label>1.0</version-label>
        <version-label>CURRENT</version-label>
    </version-labels>
    <status>INSTALLED</status>
    <type-inclusions>
        <type-inclusion>
            <type>dm_document</type>
            <include-subtypes>true</include-subtypes>
        </type-inclusion>
    </type-inclusions>
    <states>
        <state>
            <name>draft</name>
            <index>0</index>
            <no>0</no>
            <exceptional>false</exceptional>
            <allow-attach>true</allow-attach>
            <allow-schedule>true</allow-schedule>
            <allow-return-to-base>false</allow-return-to-base>
            <allow-demote>false</allow-demote>
            <return-conditions>
                <return-condition>ALL</return-condition>
            </return-conditions>
        </state>
        <state>
            <name>review</name>
            <index>1</index>
            <no>1</no>
            <exceptional>false</exceptional>
            <exception-state>fail</exception-state>
            <allow-attach>false</allow-attach>
        </state>
    </states>

```

```
<allow-schedule>true</allow-schedule>
<allow-return-to-base>true</allow-return-to-base>
<allow-demote>true</allow-demote>
<return-conditions>
    <return-condition>CHECKIN</return-condition>
    <return-condition>SAVE_AS_NEW</return-condition>
    <return-condition>BRANCH</return-condition>
</return-conditions>
<entry-criteria>
    <id>550000058000093b</id>
    <expression>
        (((subject = "ok") Or (subject = "action fail")) Or (subject = "post
fail"))
    </expression>
</entry-criteria>
<action>
    <id>0800000580004e05</id>
    <name>review_fmuzh1p0kpq8ilz5ifvexz actionset procedure for policy
        a_docbasic_lifecycle
    </name>
</action>
</state>
<state>
    <name>publish</name>
    <index>2</index>
    <no>2</no>
    <exceptional>false</exceptional>
    <allow-attach>false</allow-attach>
    <allow-schedule>true</allow-schedule>
    <allow-return-to-base>true</allow-return-to-base>
    <allow-demote>true</allow-demote>
    <return-conditions>
        <return-condition>BRANCH</return-condition>
    </return-conditions>
<entry-criteria>
    <id>550000058000093c</id>
    <expression>
        (((subject = "ok") Or (subject = "action fail")) Or (subject = "post
fail"))
    </expression>
</entry-criteria>
<action>
    <id>0800000580004e06</id>
    <name>publish_2cp0x50b4s+anynusyzqm5 actionset procedure for policy
        a_docbasic_lifecycle
    </name>
</action>
</state>
<state>
    <name>fail</name>
    <index>3</index>
    <no>3</no>
    <exceptional>true</exceptional>
    <allow-attach>false</allow-attach>
    <allow-schedule>true</allow-schedule>
    <allow-return-to-base>false</allow-return-to-base>
    <allow-demote>false</allow-demote>
    <return-conditions>
        <return-condition>ALL</return-condition>
    </return-conditions>
</state>
</states>
```

```

<links>
  <link href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
        4600000580004dfb" rel="self"/>
  <link href="http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
        rel="http://identifiers.emc.com/linkrel/owner"/>
  <link href="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document"
        rel="http://identifiers.emc.com/linkrel/type"
        title="dm_document"/>
</links>
</lifecycle>

```



### ➡ Example 2-151: JSON response

A Lifecycle sample implemented in Java.

```
{
  "id" : "4600000580004dfc",
  "name" : "a_java_lifecycle",
  "owner" : "dmadmin",
  "created" : "2017-05-03T02:15:22.000+00:00",
  "modified" : "2017-05-03T02:15:24.000+00:00",
  "implementation-type" : "JAVA",
  "version-labels" : [
    "1.0",
    "CURRENT"
  ],
  "status" : "INSTALLED",
  "type-inclusions" : [
    {
      "type" : "dm_document",
      "include-subtypes" : true
    }
  ],
  "user-validation-service" : {
    "id" : "0b00000580004df1",
    "name" : "lifecyclemodule",
    "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
  },
  "states" : [
    {
      "name" : "draft",
      "index" : 0,
      "no" : 0,
      "exceptional" : false,
      "allow-attach" : true,
      "allow-schedule" : true,
      "allow-return-to-base" : false,
      "allow-demote" : false,
      "return-conditions" : [
        "ALL"
      ]
    },
    {
      "name" : "review",
      "index" : 1,
      "no" : 1,
      "exceptional" : true,
      "allow-attach" : true,
      "allow-schedule" : true,
      "allow-return-to-base" : false,
      "allow-demote" : false,
      "return-conditions" : [
        "ALL"
      ]
    }
  ]
}
```

```
    "exceptional" : false,
    "exception-state" : "fail",
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : true,
    "allow-demote" : true,
    "return-conditions" : [
        "CHECKIN",
        "SAVE_AS_NEW",
        "BRANCH"
    ],
    "user-criteria-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    },
    "user-action-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    },
    "user-post-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    }
},
{
    "name" : "publish",
    "index" : 2,
    "no" : 2,
    "exceptional" : false,
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : true,
    "allow-demote" : true,
    "return-conditions" : [
        "BRANCH"
    ],
    "user-criteria-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    },
    "user-action-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    },
    "user-post-service" : {
        "id" : "0b00000580004df1",
        "name" : "lifecyclemodule",
        "primary-
class": "com.emc.documentum.emc.dfc.lifecycle.RestLifecycleUserProcessor"
    }
},
```

```

        "name" : "fail",
        "index" : 3,
        "no" : 3,
        "exceptional" : true,
        "allow-attach" : false,
        "allow-schedule" : true,
        "allow-return-to-base" : false,
        "allow-demote" : false,
        "return-conditions" : [
            "ALL"
        ]
    }
],
"links" : [
{
    "rel" : "self",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
        4600000580004dfc"
},
{
    "rel" : "http://identifiers.emc.com/linkrel/owner",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
},
{
    "rel" : "http://identifiers.emc.com/linkrel/type",
    "title" : "dm_document",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document"
}
]
}

```

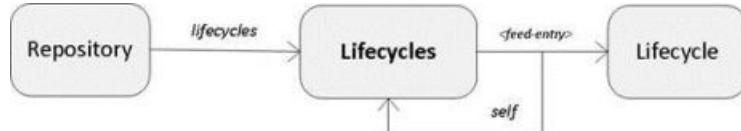


## 2.39.2 Lifecycles

The `Lifecycles` resource is used to retrieve the Lifecycles (policies) defined in the repository.

### 2.39.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagrams on page 13.

### 2.39.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Lifecycles
- Feed Updated: Server's current time
- Supports POST: No
- Entry: A single Lifecycle
- Entry ID: The Lifecycle URI
- Entry Title: The Lifecycle name
- Entry Updated: The last modify date

### 2.39.2.3 Link relations

Link relation	Description	Resource reference
self	This Lifecycles resource	<a href="#">"Lifecycles" on page 385</a>
first, last, next, previous	Pagination links	<a href="#">"Lifecycles" on page 385</a>

### 2.39.2.4 Operations

#### 2.39.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a list of Lifecycles from a repository.

#### 2.39.2.4.2 Get all Lifecycles

Retrieves a list of lifecycles from a repository.

##### 2.39.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- sort
- page
- items-per-page
- include-total
- links

- filter
- object-id: This query parameter is a string containing the object id and its default value is null.
- object-type: This query parameter is a string containing the object type name and its default value is null.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

The view parameter is not supported. When the filter parameter is specified, the client must provide the property name of the DCTM Lifecycle (policy) object, such as object\_name, or r\_object\_id.



### Caution

Providing the object-id and the object-type at the same time causes an exception to be thrown. When the object-id or the object-type are specified, only those lifecycles that are allowed to be attached by the specified object or object type, are returned. The filter parameter, in this case, is ignored.

#### 2.39.2.4.2.2 Request media types

N/A

#### 2.39.2.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.39.2.4.2.4 Request body

N/A

#### 2.39.2.4.2.5 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.39.2.4.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.39.2.4.2.7 Response status

- 200 - List of Lifecycles retrieved successfully.
- 400, 401, 403, 404, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.39.2.4.2.8 Response body

##### Example 2-152: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/lifecycles</id>
    <title>Lifecycles</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2017-05-05T09:01:47.253+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        100
    </dm:items-per-page>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles"
          rel="self"/>
    <entry>
        <id>
            http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfb
        </id>
        <title>a_docbasic_lifecycle</title>
        <author>
            <name>dmadmin</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/dmadmin</uri>
        </author>
        <updated>2017-05-03T02:15:25.000+00:00</updated>
        <published>2017-05-03T02:15:21.000+00:00</published>
        <link href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfb" rel="edit"/>
        <content src="http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000580004dfb" type="application/xml"/>
    </entry>
    <entry>
        <id>
            http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfc
        </id>
```

```

</id>
<title>a_java_lifecycle</title>
<author>
    <name>dmadmin</name>
    <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin</uri>
</author>
<updated>2017-05-03T02:15:24.000+00:00</updated>
<published>2017-05-03T02:15:22.000+00:00</published>
<link href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfc" rel="edit"/>
<content src="http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000580004dfc" type="application/xml"/>
</entry>
</feed>

```



### ➡ Example 2-153: JSON response

```
{
  "id" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles",
  "title" : "Lifecycles",
  "author" : [
    {
      "name" : "OpenText Documentum"
    }
  ],
  "updated" : "2017-05-05T09:01:28.405+00:00",
  "page" : 1,
  "items-per-page" : 100,
  "links" : [
    {
      "rel" : "self",
      "href" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles"
    }
  ],
  "entries" : [
    {
      "id" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfb",
      "title" : "a_docbasic_lifecycle",
      "author" : [
        {
          "name" : "dmadmin",
          "uri" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
        }
      ],
      "updated" : "2017-05-03T02:15:25.000+00:00",
      "published" : "2017-05-03T02:15:21.000+00:00",
      "links" : [
        {
          "rel" : "edit",
          "href" : "http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000580004dfb"
        }
      ],
      "content" : {

```

```

        "type" : "application/vnd.emc.documentum+json",
        "src" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfb"
    }
},
{
    "id" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfc",
    "title" : "a_java_lifecycle",
    "author" : [
        {
            "name" : "dmadmin",
            "uri" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
        }
    ],
    "updated" : "2017-05-03T02:15:24.000+00:00",
    "published" : "2017-05-03T02:15:22.000+00:00",
    "links" : [
        {
            "rel" : "edit",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000580004dfc"
        }
    ],
    "content" : {
        "type" : "application/vnd.emc.documentum+json",
        "src" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfc"
    }
}
]
}

```



## Caution

When getting an inline lifecycle, the following information is not returned:

- The id and primary-class of the user-validation-service
- The name of the app-validation
- The expression of entry-criteria in each state when present
- The name of the action, user-criteria, user-action, and the user-post-action in each state when present
- The id and primary-class of the user-criteria-service, the user-action-service, the user-post-service, and the system-action in each state when present

The above information is available when using the single “[Lifecycle](#)” on page 370 resource.

### 2.39.2.4.3 Additional examples for response

➡ Example 2-154: XML inline response

```

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/lifecycles</id>
  <title>Lifecycles</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2017-05-05T09:08:44.905+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
  </dm:items-per-page>
  <link
    href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles?
  inline=true"
    rel="self" />
  <entry>
    <id>
      http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
      4600000580004dfb</id>
      <title>a_docbasic_lifecycle</title>
      <author>
        <name>dmadmin</name>
        <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/dmadmin</
        uri>
      </author>
      <updated>2017-05-03T02:15:25.000+00:00</updated>
      <published>2017-05-03T02:15:21.000+00:00</published>
      <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
        4600000580004dfb"
        rel="edit" />
      <content>
        <dm:lifecycle xmlns:dm="http://identifiers.emc.com/vocab/documentum">
          <dm:id>4600000580004dfb</dm:id>
          <dm:name>a_docbasic_lifecycle</dm:name>
          <dm:owner>dmadmin</dm:owner>
          <dm:created>2017-05-03T02:15:21.000+00:00</dm:created>
          <dm:modified>2017-05-03T02:15:25.000+00:00</dm:modified>
          <dm:implementation-type>DOCBASIC</dm:implementation-type>
          <dm:version-labels>
            <dm:version-label>1.0</dm:version-label>
            <dm:version-label>CURRENT</dm:version-label>
          </dm:version-labels>
          <dm:status>INSTALLED</dm:status>
          <dm:type-inclusions>
            <dm:type-inclusion>
              <dm:type>dm_document</dm:type>
              <dm:include-subtypes>true</dm:include-subtypes>
            </dm:type-inclusion>
          </dm:type-inclusions>
          <dm:states>
            <dm:state>
              <dm:name>draft</dm:name>
              <dm:index>0</dm:index>
              <dm:no>0</dm:no>
              <dm:exceptional>false</dm:exceptional>
            </dm:state>
          </dm:states>
        </dm:lifecycle>
      </content>
    </entry>
  </feed>

```

```
<dm:allow-attach>true</dm:allow-attach>
<dm:allow-schedule>true</dm:allow-schedule>
<dm:allow-return-to-base>false</dm:allow-return-to-base>
<dm:allow-demote>false</dm:allow-demote>
<dm:return-conditions>
    <dm:return-condition>ALL</dm:return-condition>
</dm:return-conditions>
</dm:state>
<dm:state>
    <dm:name>review</dm:name>
    <dm:index>1</dm:index>
    <dm:no>1</dm:no>
    <dm:exceptional>false</dm:exceptional>
    <dm:exception-state>fail</dm:exception-state>
    <dm:allow-attach>false</dm:allow-attach>
    <dm:allow-schedule>true</dm:allow-schedule>
    <dm:allow-return-to-base>true</dm:allow-return-to-base>
    <dm:allow-demote>true</dm:allow-demote>
    <dm:return-conditions>
        <dm:return-condition>CHECKIN</dm:return-condition>
        <dm:return-condition>SAVE_AS_NEW</dm:return-condition>
        <dm:return-condition>BRANCH</dm:return-condition>
    </dm:return-conditions>
    <dm:entry-criteria>
        <dm:id>550000058000093b</dm:id>
    </dm:entry-criteria>
    <dm:action>
        <dm:id>0800000580004e05</dm:id>
    </dm:action>
</dm:state>
<dm:state>
    <dm:name>publish</dm:name>
    <dm:index>2</dm:index>
    <dm:no>2</dm:no>
    <dm:exceptional>false</dm:exceptional>
    <dm:allow-attach>false</dm:allow-attach>
    <dm:allow-schedule>true</dm:allow-schedule>
    <dm:allow-return-to-base>true</dm:allow-return-to-base>
    <dm:allow-demote>true</dm:allow-demote>
    <dm:return-conditions>
        <dm:return-condition>BRANCH</dm:return-condition>
    </dm:return-conditions>
    <dm:entry-criteria>
        <dm:id>550000058000093c</dm:id>
    </dm:entry-criteria>
    <dm:action>
        <dm:id>0800000580004e06</dm:id>
    </dm:action>
</dm:state>
<dm:state>
    <dm:name>fail</dm:name>
    <dm:index>3</dm:index>
    <dm:no>3</dm:no>
    <dm:exceptional>true</dm:exceptional>
    <dm:allow-attach>false</dm:allow-attach>
    <dm:allow-schedule>true</dm:allow-schedule>
    <dm:allow-return-to-base>false</dm:allow-return-to-base>
    <dm:allow-demote>false</dm:allow-demote>
    <dm:return-conditions>
        <dm:return-condition>ALL</dm:return-condition>
    </dm:return-conditions>
</dm:state>
```

```

        </dm:states>
        <dm:links>
            <dm:link
                href="http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
                4600000580004dfb"
                rel="self"/>
            <dm:link
                href="http://localhost:8080/dctm-rest/repositories/REPO/
users/dmadmin"
                rel="http://identifiers.emc.com/linkrel/owner"/>
            <dm:link
                href="http://localhost:8080/dctm-rest/repositories/REPO/
types/
                dm_document"
                rel="http://identifiers.emc.com/linkrel/type"
                title="dm_document"/>
        </dm:links>
    </dm:lifecycle>
</content>
</entry>
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000580004dfc</id>
    <title>a_java_lifecycle</title>
    <author>
        <name>dmadmin</name>
        <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/dmadmin</
uri>
    </author>
    <updated>2017-05-03T02:15:24.000+00:00</updated>
    <published>2017-05-03T02:15:22.000+00:00</published>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
        4600000580004dfc"
        rel="edit"/>
    <content>
        <dm:lifecycle xmlns:dm="http://identifiers.emc.com/vocab/documentum">
            <dm:id>4600000580004dfc</dm:id>
            <dm:name>a_java_lifecycle</dm:name>
            <dm:owner>dmadmin</dm:owner>
            <dm:created>2017-05-03T02:15:22.000+00:00</dm:created>
            <dm:modified>2017-05-03T02:15:24.000+00:00</dm:modified>
            <dm:implementation-type>JAVA</dm:implementation-type>
            <dm:version-labels>
                <dm:version-label>1.0</dm:version-label>
                <dm:version-label>CURRENT</dm:version-label>
            </dm:version-labels>
            <dm:status>INSTALLED</dm:status>
            <dm:type-inclusions>
                <dm:type-inclusion>
                    <dm:type>dm_document</dm:type>
                    <dm:include-subtypes>true</dm:include-subtypes>
                </dm:type-inclusion>
            </dm:type-inclusions>
            <dm:user-validation-service>
                <dm:name>lifecyclemodule</dm:name>
            </dm:user-validation-service>
            <dm:states>
                <dm:state>
                    <dm:name>draft</dm:name>
                    <dm:index>0</dm:index>

```

```
<dm:exceptional>false</dm:exceptional>
<dm:allow-attach>true</dm:allow-attach>
<dm:allow-schedule>true</dm:allow-schedule>
<dm:allow-return-to-base>false</dm:allow-return-to-base>
<dm:allow-demote>false</dm:allow-demote>
<dm:return-conditions>
    <dm:return-condition>ALL</dm:return-condition>
</dm:return-conditions>
</dm:state>
<dm:state>
    <dm:name>review</dm:name>
    <dm:index>1</dm:index>
    <dm:no>1</dm:no>
    <dm:exceptional>false</dm:exceptional>
    <dm:exception-state>fail</dm:exception-state>
    <dm:allow-attach>false</dm:allow-attach>
    <dm:allow-schedule>true</dm:allow-schedule>
    <dm:allow-return-to-base>true</dm:allow-return-to-base>
    <dm:allow-demote>true</dm:allow-demote>
    <dm:return-conditions>
        <dm:return-condition>CHECKIN</dm:return-condition>
        <dm:return-condition>SAVE_AS_NEW</dm:return-condition>
        <dm:return-condition>BRANCH</dm:return-condition>
    </dm:return-conditions>
    <dm:user-criteria-service>
        <dm:name>lifecyclemodule</dm:name>
    </dm:user-criteria-service>
    <dm:user-action-service>
        <dm:name>lifecyclemodule</dm:name>
    </dm:user-action-service>
    <dm:user-post-service>
        <dm:name>lifecyclemodule</dm:name>
    </dm:user-post-service>
</dm:state>
<dm:state>
    <dm:name>publish</dm:name>
    <dm:index>2</dm:index>
    <dm:no>2</dm:no>
    <dm:exceptional>false</dm:exceptional>
    <dm:allow-attach>false</dm:allow-attach>
    <dm:allow-schedule>true</dm:allow-schedule>
    <dm:allow-return-to-base>true</dm:allow-return-to-base>
    <dm:allow-demote>true</dm:allow-demote>
    <dm:return-conditions>
        <dm:return-condition>BRANCH</dm:return-condition>
    </dm:return-conditions>
    <dm:user-criteria-service>
        <dm:name>lifecyclemodule</dm:name>
    </dm:user-criteria-service>
    <dm:user-action-service>
        <dm:name>lifecyclemodule</dm:name>
    </dm:user-action-service>
    <dm:user-post-service>
        <dm:name>lifecyclemodule</dm:name>
    </dm:user-post-service>
</dm:state>
<dm:state>
    <dm:name>fail</dm:name>
    <dm:index>3</dm:index>
    <dm:no>3</dm:no>
    <dm:exceptional>true</dm:exceptional>
```

```

<dm:allow-attach>false</dm:allow-attach>
<dm:allow-schedule>true</dm:allow-schedule>
<dm:allow-return-to-base>false</dm:allow-return-to-base>
<dm:allow-demote>false</dm:allow-demote>
<dm:return-conditions>
    <dm:return-condition>ALL</dm:return-condition>
</dm:return-conditions>
</dm:state>
</dm:states>
<dm:links>
    <dm:link
        href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
            4600000580004dfc"
        rel="self"/>
    <dm:link
        href="http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
        rel="http://identifiers.emc.com/linkrel/owner"/>
    <dm:link
        href="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document"
        rel="http://identifiers.emc.com/linkrel/type"
title="dm_document"/>
    </dm:links>
</dm:lifecycle>
</content>
</entry>
</feed>

```



#### ➡ Example 2-155: JSON inline response

```
{
    "id" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles",
    "title" : "Lifecycles",
    "author" : [
        {
            "name" : "OpenText Documentum"
        }
    ],
    "updated" : "2017-05-05T09:09:10.214+00:00",
    "page" : 1,
    "items-per-page" : 100,
    "links" : [
        {
            "rel" : "self",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles?
inline=true"
        }
    ],
    "entries" : [
        {
            "id" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
                4600000580004dfb",
            "title" : "a_docbasic_lifecycle",
            "author" : [
                {
                    "name" : "dmadmin",
                    "uri" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
                }
            ]
        }
    ]
}
```

```
        }
    ],
    "updated" : "2017-05-03T02:15:25.000+00:00",
    "published" : "2017-05-03T02:15:21.000+00:00",
    "links" : [
        {
            "rel" : "edit",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
                4600000580004dfb"
        }
    ],
    "content" : {
        "id" : "4600000580004dfb",
        "name" : "a_docbasic_lifecycle",
        "owner" : "dmadmin",
        "created" : "2017-05-03T02:15:21.000+00:00",
        "modified" : "2017-05-03T02:15:25.000+00:00",
        "implementation-type" : "DOCBASIC",
        "version-labels" : [
            "1.0",
            "CURRENT"
        ],
        "status" : "INSTALLED",
        "type-inclusions" : [
            {
                "type" : "dm_document",
                "include-subtypes" : true
            }
        ],
        "states" : [
            {
                "name" : "draft",
                "index" : 0,
                "no" : 0,
                "exceptional" : false,
                "allow-attach" : true,
                "allow-schedule" : true,
                "allow-return-to-base" : false,
                "allow-demote" : false,
                "return-conditions" : [
                    "ALL"
                ]
            },
            {
                "name" : "review",
                "index" : 1,
                "no" : 1,
                "exceptional" : false,
                "exception-state" : "fail",
                "allow-attach" : false,
                "allow-schedule" : true,
                "allow-return-to-base" : true,
                "allow-demote" : true,
                "return-conditions" : [
                    "CHECKIN",
                    "SAVE_AS_NEW",
                    "BRANCH"
                ],
                "entry-criteria" : {
                    "id" : "550000058000093b"
                },
            }
        ]
    }
}
```

```

        "action" : {
            "id" : "0800000580004e05"
        }
    },
    {
        "name" : "publish",
        "index" : 2,
        "no" : 2,
        "exceptional" : false,
        "allow-attach" : false,
        "allow-schedule" : true,
        "allow-return-to-base" : true,
        "allow-demote" : true,
        "return-conditions" : [
            "BRANCH"
        ],
        "entry-criteria" : {
            "id" : "550000058000093c"
        },
        "action" : {
            "id" : "0800000580004e06"
        }
    },
    {
        "name" : "fail",
        "index" : 3,
        "no" : 3,
        "exceptional" : true,
        "allow-attach" : false,
        "allow-schedule" : true,
        "allow-return-to-base" : false,
        "allow-demote" : false,
        "return-conditions" : [
            "ALL"
        ]
    }
],
"links" : [
    {
        "rel" : "self",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/4600000580004dfb"
    },
    {
        "rel" : "http://identifiers.emc.com/linkrel/owner",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/users/dmadmin"
    },
    {
        "rel" : "http://identifiers.emc.com/linkrel/type",
        "title" : "dm_document",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document"
    }
]
},
{
    "id" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/4600000580004dfc",
    "title" : "a_java_lifecycle",

```

```
    "author" : [
      {
        "name" : "dmadmin",
        "uri" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dmadmin"
      }
    ],
    "updated" : "2017-05-03T02:15:24.000+00:00",
    "published" : "2017-05-03T02:15:22.000+00:00",
    "links" : [
      {
        "rel" : "edit",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000580004dfc"
      }
    ],
    "content" : {
      "id" : "4600000580004dfc",
      "name" : "a_java_lifecycle",
      "owner" : "dmadmin",
      "created" : "2017-05-03T02:15:22.000+00:00",
      "modified" : "2017-05-03T02:15:24.000+00:00",
      "implementation-type" : "JAVA",
      "version-labels" : [
        "1.0",
        "CURRENT"
      ],
      "status" : "INSTALLED",
      "type-inclusions" : [
        {
          "type" : "dm_document",
          "include-subtypes" : true
        }
      ],
      "user-validation-service" : {
        "name" : "lifecyclemodule"
      },
      "states" : [
        {
          "name" : "draft",
          "index" : 0,
          "no" : 0,
          "exceptional" : false,
          "allow-attach" : true,
          "allow-schedule" : true,
          "allow-return-to-base" : false,
          "allow-demote" : false,
          "return-conditions" : [
            "ALL"
          ]
        },
        {
          "name" : "review",
          "index" : 1,
          "no" : 1,
          "exceptional" : false,
          "exception-state" : "fail",
          "allow-attach" : false,
          "allow-schedule" : true,
          "allow-return-to-base" : true,
          "allow-demote" : true,
        }
      ]
    }
  ]
}
```

```

    "return-conditions" : [
        "CHECKIN",
        "SAVE_AS_NEW",
        "BRANCH"
    ],
    "user-criteria-service" : {
        "name" : "lifecyclemodule"
    },
    "user-action-service" : {
        "name" : "lifecyclemodule"
    },
    "user-post-service" : {
        "name" : "lifecyclemodule"
    }
},
{
    "name" : "publish",
    "index" : 2,
    "no" : 2,
    "exceptional" : false,
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : true,
    "allow-demote" : true,
    "return-conditions" : [
        "BRANCH"
    ],
    "user-criteria-service" : {
        "name" : "lifecyclemodule"
    },
    "user-action-service" : {
        "name" : "lifecyclemodule"
    },
    "user-post-service" : {
        "name" : "lifecyclemodule"
    }
},
{
    "name" : "fail",
    "index" : 3,
    "no" : 3,
    "exceptional" : true,
    "allow-attach" : false,
    "allow-schedule" : true,
    "allow-return-to-base" : false,
    "allow-demote" : false,
    "return-conditions" : [
        "ALL"
    ]
},
],
"links" : [
    {
        "rel" : "self",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/4600000580004dfc"
    },
    {
        "rel" : "http://identifiers.emc.com/linkrel/owner",
        "href" : "http://localhost:8080/dctm-rest/repositories/REPO/users/dmadmin"
    }
]
}

```

```
        },
        {
            "rel" : "http://identifiers.emc.com/linkrel/type",
            "title" : "dm_document",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_document"
        }
    ]
}
]
```

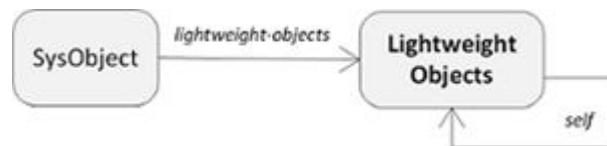


## 2.40 Lightweight Objects

The **Lightweight Objects** resource represents lightweight objects that share a shareable object.

## 2.40.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

## 2.40.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of lightweight objects.
  - Feed Title: Objects
  - Feed Updated: Server's current time
  - Supports POST: Yes
  - Entry: “**SysObject**” on page 692

### 2.40.3 Link relations

Link relation	Description	Resource reference
self	This Lightweight Objects resource	"Lightweight Objects" on page 400
first, last, next, previous	Pagination links.	"SysObject" on page 692

### 2.40.4 Operations

#### 2.40.4.1 Supported HTTP methods

Method	Description
POST	Creates a new lightweight object that shares this shareable object
GET	Retrieves all lightweight objects that share this shareable object

#### 2.40.4.2 Create a lightweight object

Create a lightweight object and share this shareable object.



**Note:** While creating the lightweight object, if a lightweight object modifies the shared properties, then the lightweight object is materialized. This mechanism of lightweight type is called as auto materialized.

One shared object can only be shared by one kind of lightweight type object at the same time.

##### 2.40.4.2.1 HTTP Request method and query parameters

POST

##### 2.40.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.40.4.2.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.40.4.2.4 Request body

##### ➡ Example 2-156: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<object>
  <properties>
    <object_name>xyz</object_name>
    <r_object_type>lightweight type</r_object_type>
    ...
  </properties>
</object>
```



##### ➡ Example 2-157: JSON request

```
{
  "name": "object",
  "properties": {
    "object_name": "xyz",
    "r_object_type": "lightweight type",
    ...
  }
}
```



#### 2.40.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.40.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.40.4.2.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.40.4.2.8 Response body

##### ➡ Example 2-158: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<object>
  <properties>
    <object_name>xyz</object_name>
    <r_object_id>090020808001398e</r_object_id>
    ...
  </properties>
  <links>
    <link rel="self" href="{objectResourceUri}" />
    ...
  </links>
</object>
```



##### ➡ Example 2-159: JSON response

```
{
  "name": "object",
  "type": "dm_sysobject",
  "definition": "",
  "properties": {
    "object_name": "xyz",
    "r_object_id": "090020808001398e",
    ...
  },
  "links": [
    { "rel": "self", "href": "objectResourceUri" },
    ...
  ]
}
```



#### 2.40.4.3 Get the Lightweight Objects of a shared Object

GET all the lightweight objects of the specified shared parent object.

##### 2.40.4.3.1 HTTP Request method and query parameters

GET

- inline
- items-per-page
- view
- include-total
- page
- sort
- filter
- links
- q



**Note:** The `view`, `sort`, and `filter` query parameters do not support the properties of the shared parent.

For example, an exception may be thrown when `view=i_folder_id&inline=true`.

The `q` parameter is supported for full text searching with a subset of the Simple Search Language, however parenthesis are not supported.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

##### 2.40.4.3.2 Request media types

N/A

##### 2.40.4.3.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.40.4.3.4 Request body

N/A

#### 2.40.4.3.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.40.4.3.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.40.4.3.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.40.4.3.8 Response body

##### Example 2-160: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0800000580005381/lw-objects</id>
    <title>Objects</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2015-12-28T07:31:31.795+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        100</dm:items-per-page>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0800000580005381/lw-objects?inline=true" rel="self"/>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0800000580005d9a</id>
        <title>new lightweight 29</title>
        <author>
            <name>dave</name>
```

```

<uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
dave</uri>
</author>
<summary>my_lightweight_object 0800000580005d9a</summary>
<updated>2015-12-28T07:00:24.000+00:00</updated>
<published>2015-12-28T07:00:24.000+00:00</published>
<link href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
0800000580005d9a" rel="edit"/>
<content>
    <dm:object definition="http://localhost:8080/dctm-rest/
repositories/REPO/
types/my_lightweight_object"
    xmlns:dm="http://identifiers.emc.com/vocab/documentum"
        xsi:type="my_lightweight_object">
        <dm:properties>
            <dm:object_name>new lightweight 29</dm:object_name>
            <dm:i_sharing_parent>0800000580005381</dm:i_sharing_parent>
            <dm:r_object_type>my_lightweight_object</dm:r_object_type>
            <dm:r_page_cnt>0</dm:r_page_cnt>
            <dm:r_object_id>0800000580005d9a</dm:r_object_id>
            <dm:r_modify_date>2015-12-28T07:00:24.000+00:00
            </dm:r_modify_date>
            <dm:owner_name>dave</dm:owner_name>
            <dm:creation-date>2015-12-28T07:00:24.000+00:00
            </dm:creation-date>
        </dm:properties>
        <dm:links>
            <link href="http://localhost:8080/dctm-rest/repositories/
REPO/
objects/0800000580005d9a" rel="self"/>
            ...
        </dm:links>
    </dm:object>
</content>
</entry>
</feed>

```



### Example 2-161: JSON response

```
{
    "id" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0800000580005381/
    lw-objects",
    "title" : "Objects",
    "author" : [
        {
            "name" : "OpenText Documentum"
        }
    ],
    "updated" : "2015-12-28T07:57:27.299+00:00",
    "page" : 1,
    "items-per-page" : 100,
    "links" : [
        {
            "rel" : "self",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0800000580005381/lw-objects?inline=true"
        }
    ],
}
```

```

    "entries" : [
      {
        "id" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0800000580005d9a",
        "title" : "new lightweight 29",
        "author" : [
          {
            "name" : "dave",
            "uri" : "http://localhost:8080/dctm-rest/repositories/REPO/users/
dave"
          }
        ],
        "summary" : "my_lightweight_object 0800000580005d9a",
        "updated" : "2015-12-28T07:00:24.000+00:00",
        "published" : "2015-12-28T07:00:24.000+00:00",
        "links" : [
          {
            "rel" : "edit",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0800000580005d9a"
          }
        ],
        "content" : {
          "name" : "object",
          "type" : "my_lightweight_object",
          "definition" : "http://localhost:8080/dctm-rest/repositories/REPO/
types/
          my_lightweight_object",
          "properties" : {
            "object_name" : "new lightweight 29",
            "i_sharing_parent" : "0800000580005381",
            "r_object_type" : "my_lightweight_object",
            "r_page_cnt" : 0,
            "r_object_id" : "0800000580005d9a",
            "r_modify_date" : "2015-12-28T07:00:24.000+00:00",
            "owner_name" : "dave",
            "creation-date" : "2015-12-28T07:00:24.000+00:00"
          },
          "links" : [
            {
              "rel" : "self",
              "href" : "http://localhost:8080/dctm-rest/repositories/REPO/
objects/
              0800000580005d9a"
            },
            ...
          ]
        }
      ]
    }
  
```

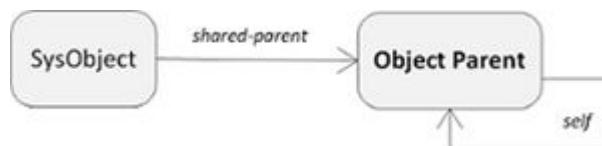


## 2.41 Lightweight Object Parent

The Lightweight Object Parent resource represents the parent sharable object of a lightweight object.

### 2.41.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

### 2.41.2 Feed

No feed support.

### 2.41.3 Link relations

Link relation	Description	Resource reference
self	This Lightweight Object Parent resource	<a href="#">“Lightweight Object Parent” on page 408</a>

This resource also has all of the same link relations as the “[SysObject](#)” on page 692 resource.

### 2.41.4 Operations

#### 2.41.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the lightweight parent shareable object
POST	Reassign the parent to another shareable object

#### 2.41.4.2 Get the Parent Shareable Object

Get the parent shareable object of a lightweight object.

##### 2.41.4.2.1 HTTP Request method and query parameters

GET

- view
- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.41.4.2.2 Request media types

N/A

##### 2.41.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.41.4.2.4 Request body

N/A

##### 2.41.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.41.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.41.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.41.4.2.8 Response body

##### Example 2-162: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<object>
  <properties>
    <object_name>xyz</object_name>
    <r_object_id>090020808001398e</r_object_id>
    ...
  </properties>
  <links>
    <link rel="self" href="{objectResourceUri}" />
    ...
  </links>
</object>
```



##### Example 2-163: JSON response

```
{
  "name": "object",
  "type": "shareable_type",
  "definition": "",
  "properties": {
    "object_name": "xyz",
    "r_object_id": "090020808001398e",
    ...
  }
  "links": [
    { "rel": "self", "href": "objectResourceUri" },
    ...
  ]
}
```



### 2.41.4.3 Reassign the parent a lightweight object

Reassign the parent of the lightweight object to another shareable object.

#### 2.41.4.3.1 HTTP Request method and query parameters

POST

#### 2.41.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.41.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.41.4.3.4 Request body

► **Example 2-164: XML request**

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:object href="http://localhost/dctm-rest/repositories/acme/objects/xxxxx"/>
```



► **Example 2-165: JSON request**

```
{
  "href": "http://localhost/dctm-rest/repositories/acme/objects/xxxxx"
}
```



#### 2.41.4.3.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.41.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.41.4.3.7 Response status

- 200 - New parent successfully assigned
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.41.4.3.8 Response body

The same as the “[SysObject](#)” on page 692 resource.

##### ➡ Example 2-166: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<object xmlns="http://identifiers.emc.com/vocab/documentum"
    definition="{typeResourceUri}"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="shareable_type">
<properties>
    <object_name>xyz</object_name>
    <r_object_id>090020808001398e</r_object_id>
    ...
</properties>
<links>
    <link rel="self" href="{documentResourceUri}" />
    ...
</links>
</document>
```



##### ➡ Example 2-167: JSON response

```
{
    "name": "document",
    "type": "shareable_type",
    "definition": "<documentResourceUri>",
    "properties": {
        "object_name": "xyz",
        "r_object_id": "090020808001398e",
        ...
    }
    "links": [
        { "rel": "self", "href": "<documentResourceUri>" },
        ...
    ]
}
```



## 2.42 Lightweight Object Materialization

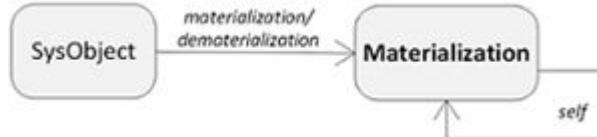
The Lightweight Object Materialization resource represents the materialization of a lightweight object.

### 2.42.1 Feed

No feed support.

### 2.42.2 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

### 2.42.3 Link relations

Link relation	Description	Resource reference
self	This Lightweight Object Parent resource	<a href="#">“Lightweight Object Materialization” on page 413</a>
The same relations as “ <a href="#">SysObject</a> ” on page 692.		

### 2.42.4 Operations

#### 2.42.4.1 Supported HTTP methods

Method	Description
PUT	Materialize the lightweight object
DELETE	Dematerialize the lightweight object

#### **2.42.4.2 Materialize the lightweight object**

Materializes an unmaterialized lightweight object.

##### **2.42.4.2.1 HTTP Request method and query parameters**

PUT

##### **2.42.4.2.2 Request media types**

N/A

##### **2.42.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### **2.42.4.2.4 Request body**

N/A

##### **2.42.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### **2.42.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

##### **2.42.4.2.7 Response status**

- 200 - Materialized successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.42.4.2.8 Response body

➡ Example 2-168: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<object>
<properties>
    <object_name>xyz</object_name>
    <r_object_id>090020808001398e</r_object_id>
    ...
</properties>
<links>
    <link rel="self" href="{objectResourceUri}" />
    <link rel="http://identifiers.emc.com/linkrel/dematerialize" href="" />
    <link rel="http://identifiers.emc.com/linkrel/parent" href="" />
    ...
</links>
</object>
```



➡ Example 2-169: JSON response

```
{
    "name": "object",
    "type": "dm_sysobject",
    "definition": "",
    "properties": {
        "object_name": "xyz",
        "r_object_id": "090020808001398e",
        ...
    },
    "links": [
        { "rel": "self", "href": "objectResourceUri" },
        { "rel": "http://identifiers.emc.com/linkrel/dematerialize", "href": "" },
        { "rel": "http://identifiers.emc.com/linkrel/parent", "href": "" },
        ...
    ]
}
```



#### 2.42.4.3 Dematerialize a lightweight object

Dematerialize a materialized lightweight object.

##### 2.42.4.3.1 HTTP Request method and query parameters

DELETE

#### **2.42.4.3.2 Request media types**

N/A

#### **2.42.4.3.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.42.4.3.4 Request body**

N/A

#### **2.42.4.3.5 Response headers**

N/A

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.42.4.3.6 Response media types**

N/A

#### **2.42.4.3.7 Response status**

- 204 - Dematerialized successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.42.4.3.8 Response body**

N/A

### **2.43 Lock**

The Lock resource represents a lock of the SysObject.

## 2.43.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

## 2.43.2 Feed

No feed support.

## 2.43.3 Link relations

The Lock resource is used as a link relation in the SysObject resource. See “[SysObject](#)” on page 692.

## 2.43.4 Operations

### 2.43.4.1 Supported HTTP methods

Method	Description
PUT	Checks out a SysObject.
DELETE	Cancels a check-out operation on a SysObject.

### 2.43.4.2 Check out (lock) an object

Check out a SysObject.

#### 2.43.4.2.1 HTTP Request method and query parameters

PUT

#### **2.43.4.2.2 Request media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### **2.43.4.2.3 Request headers**

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.43.4.2.4 Request body**

N/A

#### **2.43.4.2.5 Response headers**

- Content-Type

#### **2.43.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.43.4.2.7 Response status**

- 200 - Checked out successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.43.4.2.8 Response body**

XML or JSON representation of the object. A successful response contains all properties of the checked out object.

### 2.43.4.3 Cancel the check-out (unlock) operation on an object.

Cancel the check-out operation on a SysObject.

#### 2.43.4.3.1 HTTP Request method and query parameters

DELETE

#### 2.43.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.43.4.3.3 Request headers

- Authorization
- Accept
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.43.4.3.4 Request body

N/A

#### 2.43.4.3.5 Response headers

N/A

#### 2.43.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.43.4.3.7 Response status

- 204 - Unlocked successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.43.4.3.8 Response body

HTTP 204 No Content status upon a successful unlock operation. The Response body contains no content.

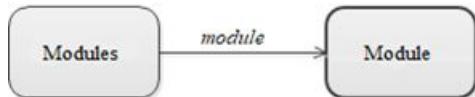
### 2.44 Module(s)

#### 2.44.1 Module

The **Module** resource represents an object instance of a module.

##### 2.44.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

##### 2.44.1.2 Feed

No feed support.

##### 2.44.1.3 Link relations

Link relation	Description	Resource reference
self	This Module resource	<a href="#">“Module” on page 420</a>
edit	Edit this Module resource	<a href="#">“Module” on page 420</a>

##### 2.44.1.4 Operations

###### 2.44.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves an instance of a givenModule.
POST	Updates an instance of a given Module.

#### 2.44.1.4.2 Get Module details

Using the specified module id, this method retrieves the specified Module instance details.

Each Module instance has five attributes:

- *name*: The name of the resource object.
- *id*: The id of the module.
- *moduleName*: The name of the module.
- *description*: A brief description of the module.
- *Status*: The current state of whether the module is enabled or disabled.

##### 2.44.1.4.2.1 HTTP Request method and query parameters

GET

##### 2.44.1.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.44.1.4.2.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.44.1.4.2.4 Request body

N/A

##### 2.44.1.4.2.5 Response headers

- Content-Type

#### 2.44.1.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json

#### 2.44.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.44.1.4.2.8 Response body

XML or JSON representation of the Module object instance. A successful Response contains all of the properties of the retrieved Module instance.

##### Example 2-170: JSON Response

```
...
{
    "name": "module",
    "id": "PRM",
    "description": "Physical Records Management provides functions to model
                    physical objects in Documentum and also provides library
                    services.",
    "status": "enabled",
    "moduleName": "Physical Records Manager",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/documentum2/modules/pRM"
        }
    ]
}
```



##### Example 2-171: Request

```
Accept: application/vnd.emc.documentum+json
GET http://localhost/dctm-rest/repositories/documentum2/modules/RPS
```



##### Example 2-172: JSON Response

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8
Date: Tue, 16 Oct 2012 01:27:01 GMT
Server: Apache-Coyote/1.1
Transfer-Encoding: chunked
```

```
{  
    "name": "module",  
    "id": "RPS",  
    "description": "Automates policy-based storage and archival of content.",  
    "status": "disabled",  
    "moduleName": "Retention Policy Services",  
    "links": [  
        {  
            "rel": "self",  
            "href": "http://localhost:8080/dctm-rest/repositories/documentum2/  
                    modules/rps"  
        },  
        {  
            "rel": "edit",  
            "href": "http://localhost:8080/dctm-rest/repositories/documentum2/  
                    modules/RPS"  
        }  
    ]  
}
```



#### 2.44.1.4.3 Update the Module status to enabled

Update the module status to the *enabled* state.

##### 2.44.1.4.3.1 HTTP Request method and query parameters

POST

##### 2.44.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.44.1.4.3.3 Request headers

- Authorization
- Accept
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.44.1.4.3.4 Request body

Module instance in XML or JSON representation.

#### 2.44.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.44.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.44.1.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.44.1.4.3.8 Response body

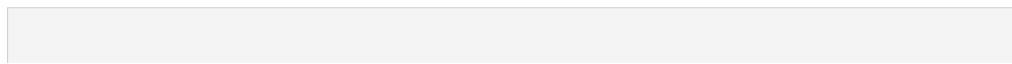
The details, in XML or JSON, of a single Module instance. A successful Response contains the attributes of the specified module, and updates the status to enabled.

##### Example 2-173: JSON Response

```
...
{
    "name": "module",
    "id": "PRM",
    "description": "Physical Records Management provides functions to model physical objects in Documentum and also provides library services.",
    "status": "enabled",
    "moduleName": "Physical Records Manager",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/documentum2/modules/pRM"
        }
    ]
}
```



##### Example 2-174: JSON Response



```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "name": "module",
  "id": "PRM",
  "description": "Physical Records Management provides functions to model
                  physical objects in Documentum and also provides library
                  services.",
  "status": "enabled",
  "moduleName": "Physical Records Manager",
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/documentum2/
               modules/pRM"
    }
  ]
}

```

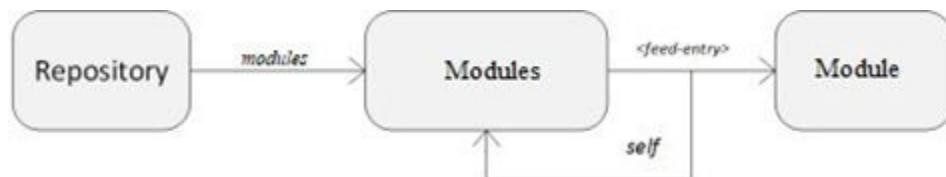


## 2.44.2 Modules

The Modules collection resource represents a collection of modules in a repository.

### 2.44.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.44.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Modules
- Feed Updated: The current time
- Entry: “Module” on page 420
- Supports POST: No

### 2.44.2.3 Link relations

Link relation	Description	Resource reference
self	The current collection of modules	<a href="#">Modules</a>

### 2.44.2.4 Operations

#### 2.44.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of modules in a repository.

#### 2.44.2.4.2 Get Modules

Retrieves a list of modules in a repository.

##### 2.44.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	inline	Determines whether to show content (the object instance) in an atom entry for a collection.	boolean	true or false  true - Returns an object instance, which is put into the entry's <i>content</i> element.  false - Does not return anything.	false

HTTP method	Query parameters	Description	Data type	Value range	Default value
	links	Determines whether a link relation is returned for this object representation.	boolean	true or false true - The default value. false - means "do not send me any links that are expensive to calculate, but it is fine if you send me ones that are cheap"	true

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.44.2.4.2.2 Request media types

N/A

#### 2.44.2.4.2.3 Request headers

- Accept
- Content-Type
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.44.2.4.2.4 Request body

N/A

#### 2.44.2.4.2.5 Response headers

- Content-Type
- Content-Length

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.44.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json

#### 2.44.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.44.2.4.2.8 Response body

An XML or JSON representation of the modules collection.

- The body contains a list of modules for a repository.

##### ➡ Example 2-175: JSON

```
1
2
3 {
4     "id": "http://localhost:8080/dctm-rest/repositories/documentum1/modules",
5     "title": "Modules",
6     "author": [
7         {
8             "name": "OpenText Documentum"
9         }
10    ],
11    "updated": "2020-02-04T10:31:38.208+00:00",
12    "page": 1,
13    "items-per-page": 1000,
14    "total": 5,
15    "links": [
16        {
17            "rel": "self",
18            "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
19                modules"
20        }
21    ],
22    "entries": [
23        {
24            "id": "http://localhost:8080/dctm-rest/repositories/documentum1/
25                modules/RM",
26            "title": "Records Manager",
27            "summary": "Enable the capability for clients to declare formal records
28                        and create relationships among these records. Allows
29                        administrators to configure and apply policies to create
30                        their own file plans.
31                        Requires RPS. - Enabled",
32            "updated": "2020-02-04T10:31:38.209+00:00",
33            "published": "2020-02-04T10:31:38.209+00:00",
34            "links": [
35                {
36                    "rel": "edit",
37                    "href": "http://localhost:8080/dctm-rest/repositories/
38                        documentum1/modules/RM"
39                }
40            ],
41            "content": {
42                "type": "application/vnd.emc.documentum+json",
43                "src": "http://localhost:8080/dctm-rest/repositories/documentum1/
44                    modules/RM"
45            }
46        },
47    ]
48}
```

```

43     ...
44   ]
45 }
```



### Example 2-176: Request

```

1
2
3 Accept: application/vnd.emc.documentum+json
4
5 GET http://localhost/dctm-rest/repositories/documentum1/modules?inline=true
```



### Example 2-177: Response

```

1
2
3 Status Code: 200 OK
4 Content - Type: application / vnd.emc.documentum + json;
5 charset = UTF - 8
6 Date: Tue, 16 Oct 2012 01: 27: 01 GMT
7 Server: Apache - Coyote / 1.1
8 Transfer - Encoding: chunked
9
10 {
11   "id": "http://localhost:8080/dctm-rest/repositories/documentum1/modules",
12   "title": "Modules",
13   "author": [
14     {
15       "name": "OpenText Documentum"
16     }
17   ],
18   "updated": "2020-02-04T09:35:39.361+00:00",
19   "page": 1,
20   "items-per-page": 1000,
21   "total": 5,
22   "links": [
23     {
24       "rel": "self",
25       "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
26           modules?inline=true"
27     }
28   ],
29   "entries": [
30     {
31       "id": "http://localhost:8080/dctm-rest/repositories/documentum1/modules/
32           RM",
33       "title": "Records Manager",
34       "summary": "Enable the capability for clients to declare formal records
35                   and create relationships among these records. Allows
36                   administrators to configure and apply policies to create
37                   their own file plans.
38                   Requires RPS. - Enabled",
39       "updated": "2020-02-04T09:35:39.363+00:00",
40       "published": "2020-02-04T09:35:39.363+00:00",
41       "links": [
42         {
43           "rel": "edit",
44           "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
45               modules/RM"
46         }
47       ],
48       "content": {
```

```
49          "name": "module",
50          "id": "RM",
51          "description": "Enable the capability for clients to declare formal
52                          records and create relationships among these records.
53                          Allows administrators to configure and apply policies
54                          to create their own file plans.Requires RPS.",
55          "status": "enabled",
56          "moduleName": "Records Manager",
57          "links": [
58              {
59                  "rel": "self",
60                  "href": "http://localhost:8080/dctm-rest/repositories/
61                      documentum1/modules/RM"
62              }
63          ]
64      },
65      {
66          "id": "http://localhost:8080/dctm-rest/repositories/documentum1/modules/
67              RPS",
68          "title": "Retention Policy Services",
69          "summary": "Automates policy-based storage and archival of
70                          content. - Enabled",
71          "updated": "2020-02-04T09:35:39.363+00:00",
72          "published": "2020-02-04T09:35:39.363+00:00",
73          "links": [
74              {
75                  "rel": "edit",
76                  "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
77                      modules/RPS"
78              }
79          ],
80          "content": {
81              "name": "module",
82              "id": "RPS",
83              "description": "Automates policy-based storage and archival of
84                          content.",
85              "status": "enabled",
86              "moduleName": "Retention Policy Services",
87              "links": [
88                  {
89                      "rel": "self",
90                      "href": "http://localhost:8080/dctm-rest/repositories/
91                          documentum1/modules/RPS"
92                  }
93              ]
94          }
95      },
96      {
97          "id": "http://localhost:8080/dctm-rest/repositories/documentum1/modules/
98              PRM",
99          "title": "Physical Records Manager",
100         "summary": "Physical Records Management provides functions to model
101                         physical objects in Documentum and also provides library
102                         services. - Enabled",
103         "updated": "2020-02-04T09:35:39.363+00:00",
104         "published": "2020-02-04T09:35:39.363+00:00",
105         "links": [
106             {
107                 "rel": "edit",
108                 "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
109                     modules/PRM"
110             }
111         ],
112         "content": {
113             "name": "module",
114             "id": "PRM",
115             "description": "Physical Records Management provides functions to
116                         model physical objects in Documentum and also
117                         provides library services.",
```

```

119         "status": "enabled",
120         "moduleName": "Physical Records Manager",
121         "links": [
122             {
123                 "rel": "self",
124                 "href": "http://localhost:8080/dctm-rest/repositories/
125                     documentum1/modules/PRM"
126             }
127         ]
128     },
129     {
130         "id": "http://localhost:8080/dctm-rest/repositories/documentum1/modules/
131             FRS",
132         "title": "Federated Record Services",
133         "summary": "Manages records, valuable and discoverable corporate assets,
134             across the enterprise or multiple repositories. - Enabled",
135         "updated": "2020-02-04T09:35:39.364+00:00",
136         "published": "2020-02-04T09:35:39.364+00:00",
137         "links": [
138             {
139                 "rel": "edit",
140                 "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
141                     modules/FRS"
142             }
143         ],
144         "content": {
145             "name": "module",
146             "id": "FRS",
147             "description": "Manages records, valuable and discoverable corporate
148                 assets, across the enterprise or multiple
149                 repositories.",
150             "status": "enabled",
151             "moduleName": "Federated Record Services",
152             "links": [
153                 {
154                     "rel": "self",
155                     "href": "http://localhost:8080/dctm-rest/repositories/
156                         documentum1/modules/FRS"
157                 }
158             ]
159         }
160     },
161     {
162         "id": "http://localhost:8080/dctm-rest/repositories/documentum1/
163             modules/Collaboration",
164         "title": "Collaborative Edition",
165         "summary": "Provides Web-based collaborative workplace. - Enabled",
166         "updated": "2020-02-04T09:35:39.364+00:00",
167         "published": "2020-02-04T09:35:39.364+00:00",
168         "links": [
169             {
170                 "rel": "edit",
171                 "href": "http://localhost:8080/dctm-rest/repositories/documentum1/
172                     modules/Collaboration"
173             }
174         ],
175         "content": {
176             "name": "module",
177             "id": "Collaboration",
178             "description": "Provides Web-based collaborative workplace.",
179             "status": "enabled",
180             "moduleName": "Collaborative Edition",
181             "links": [
182                 {
183                     "rel": "self",
184                     "href": "http://localhost:8080/dctm-rest/repositories/
185                         documentum1/modules/Collaboration"
186                 }
187             ]
188         }

```

```
189 }  
190 }  
191 ]  
192 }
```



## 2.45 Multiple Repository Search

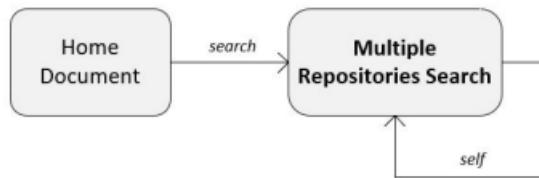
The *Multiple Repository Search* resource allows you to define multiple repositories as the scope of a search and obtain search results from all of those repositories with one HTTP request.



**Note:** Currently, only repositories from Documentum CM Server are supported.

### 2.45.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Home Document” on page 333 and About the Diagram on page 13.

### 2.45.2 Feed

This resource is a feed that wraps the search results returned from all repositories that were specified by the client in the search criteria. Each entry represents an object within one of the repositories.

- Feed ID: Resource URI
- Feed Title: Search results of the repositories
- Feed Updated: Current server time
- Supports POST: Yes
- Entry: Search results
- Entry ID: The object id of an object returned in the search results
- Entry Title: The object name of an object returned in the search results
- Entry Updated: Current server time

### 2.45.3 Link relations

Link relation	Description	Resource reference
self	The URI of the current search	"Multiple Repository Search" on page 432
search [1]	The URI template for any subsequent searches	"Multiple Repository Search" on page 432
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		



**Note:** Since pagination is not effective in this resource, pagination link relations, such as *previous*, *last*, *first*, and *next*, are not supported.

### 2.45.4 Operations

#### 2.45.4.1 Result range

This resource does not use any pagination parameters, such as `page` and `items-per-page`, to determine the range of the search in the way that the Single Search Repository resource does. Instead, the `start` and `limit` URI parameters are used to define the range of the search for each repository separately. For example, if you have REPO1 and REPO2 (assuming that they produce a sufficient number of search results), and you use the `start=1&limit=10` URI parameters for each of those repositories. Using these URI parameters returns a total of 20 search results in the Response. In this case, there are 10 results from REPO1 and 10 results from REPO2 that are returned in the Response.

#### 2.45.4.2 Supported HTTP methods

Method	Description
GET	Use URI parameters to search across many repositories
POST	Use AQL and URI parameters to search across many repositories

#### 2.45.4.3 Search by Get HTTP method

Search all repositories with global login information using the GET HTTP method. Searching using the GET method does not allow you to choose the repositories to be searched because all repository names that are available are cached after REST has retrieved them for the first time.

If some of the repositories change, the cached repository names will become obsolete after a certain amount of time. Detailed cache properties can be found and customized (by overriding the default ones) in the `rest-api-common-ehcache.xml` file, which is located at `<dctm-rest.war>/WEB-INF/classes/`.

##### 2.45.4.3.1 HTTP Request method and query parameters

This resource uses the GET HTTP method and the following URI parameters. Most URI parameters have the same usage as those of the [Search](#) resource. However, some URI parameters are not supported.

The following table shows the details of the URI parameters:

Parameter Type	Parameter	Support	Description
Common URI parameters	include-total	Yes	When only one repository is being searched, the <i>total</i> field is returned.  When more than one repository is being searched, a Map named <i>total-for-each</i> is returned rather than the <i>total</i> field.
	inline	Yes	
	page	No	An exception is thrown when this parameter is specified.
	items-per-page	No	An exception is thrown when this parameter is specified.
	start	Yes	Defines the start index of the search range for each repository.  The default value is 1.
	limit	Yes	Defines the maximum size of the search range for each repository.  The default value is 100.
	sort	Yes	Sorts the search results of each repository.

Parameter Type	Parameter	Support	Description
	view	Yes	<ul style="list-style-type: none"> <li>When no value for this <i>view</i> parameter is specified, the <i>:default</i> attributes are: <ul style="list-style-type: none"> <li>r_object_id</li> <li>object_name</li> <li>r_object_type</li> <li>r_modify_date</li> <li>r_modifier</li> <li>summary</li> </ul> </li> <li><i>:all</i> not supported as it is the same as <i>:default</i></li> <li>Custom view is supported and it overwrites the attributes of <i>:default</i>.  <b>view=r_object_id,r_object_type&amp;inline=true</b>  This returns two attributes</li> </ul>
Search URI parameters	q	Yes	A full-text expression as defined in Simple Search Language.
	locations	No	A list of folder paths. The query is restricted to these folders.
	collections	Yes	A list of collection names. These names must match the collections defined in xPlore.
	object-type	Yes	Target object type to search against.
	facet	Yes	Facet is applied across all repositories. A general count of the search results for all of the repositories is returned.
	facet-value-constraints	Yes	Constraints attributes for the <i>facet</i> parameter. The format is <i>facet-value-constraints=attr_constraint_expression</i>
	query-locale	Yes	A locale used to evaluate the query and parse words into a language dependent token. Only supported with <i>xPlore</i> . The default value is <i>en</i> .

#### **2.45.4.3.2 Request media types**

N/A

#### **2.45.4.3.3 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.45.4.3.4 Request body**

N/A

#### **2.45.4.3.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.45.4.3.6 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### **2.45.4.3.7 Response status**

- 200 - Retrieved successfully
- 201,204,400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.45.4.3.8 Response body

➡ Example 2-178: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/search</id>
    <title>Search results</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2018-12-07T06:26:06.665+00:00</updated>
    <dm:total-for-each xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        <dm:REPO>0</dm:REPO>
        <dm:REPO1>2</dm:REPO1>
    </dm:total-for-each>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/search.xml?start=1&limit=1">
    </link>
    <link rel="search"
          hreftemplate="http://localhost:8080/dctm-rest/search.xml{?
                        collections,facet,include-total,inline,limit,locations,
                        object-type,q,start,timezone,view}"></link>
    <entry>
        <id>0c00000280002104</id>
        <title>auditadmin1</title>
        <author>
            <name>dmadmin1</name>
        </author>
        <updated>2018-06-13T03:20:15.000+00:00</updated>
        <link rel="edit"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/objects/
                    0c00000280002104.xml"></link>
        <content type="application/xml"
                 src="http://localhost:8080/dctm-rest/repositories/REPO1/objects/
                     0c00000280002104.xml"></content>
        <relevance:score xmlns:relevance="http://a9.com/-/opensearch/extensions/
                           relevance/1.0/">1.0</relevance:score>
        <dm:source xmlns:dm="http://identifiers.emc.com/vocab/documentum">
            REPO1
        </dm:source>
    </entry>
</feed>
```



➡ Example 2-179: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/search",
    "title": "Search results",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-12-07T06:27:02.819+00:00",
    "total-for-each": {
        "REPO": 0,
        "REPO1": 2
    },
}
```

```

    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/search?start=1&limit=1"
        },
        {
            "rel": "search",
            "hreftemplate": "http://localhost:8080/dctm-rest/search{?collections,
                           facet,include-total,inline,limit,locations,
                           object-type,q,start,timezone,view}"
        }
    ],
    "entries": [
        {
            "id": "0c00000280002104",
            "title": "auditadmin1",
            "author": [
                {
                    "name": "dmadmin1"
                }
            ],
            "updated": "2018-06-13T03:20:15.000+00:00",
            "links": [
                {
                    "rel": "edit",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
                           objects/0c00000280002104"
                }
            ],
            "content": {
                "type": "application/vnd.emc.documentum+json",
                "src": "http://localhost:8080/dctm-rest/repositories/REPO1/objects/
                       0c00000280002104"
            },
            "score": "1.0",
            "source": "REPO1"
        }
    ]
}

```



### Example 2-180: JSON + HAL response

```

{
    "id": "http://localhost:8080/dctm-rest/search",
    "title": "Search results",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-12-10T07:37:09.054+00:00",
    "_embedded": {
        "collection": [
            {
                "id": "0900000280001629",
                "title": "Doc-3.pdf",
                "author": [
                    {
                        "name": "Administrator"
                    }
                ],
                "summary": "Manage Document Versions in REST Services. This doc
                           shows a series of JSON message samples to do
                           versioning in Documentum REST Services. Get a document
                           resource Assumed that you get a document resource.
                           GET http://localhost:8080/dctm-rest/repositories/rep",
                "updated": "2018-11-09T12:52:14.000+00:00",
            }
        ]
    }
}

```

```

        "score": "0.5375551581382751",
        "terms": [
            "java"
        ],
        "source": "REPO1",
        "_links": {
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
objects/0900000280001629.hal"
            }
        }
    ],
    "_links": {
        "search": {
            "href": "http://localhost:8080/dctm-rest/search.hal{?collections,
facet,include-total,inline,limit,locations,object-type,q,
start,timezone,view}",
            "templated": true
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/search.hal?q=java"
        }
    }
}

```



#### 2.45.4.4 Search by POST HTTP method

The POST HTTP method allows you to search across multiple repositories using AQL and URI parameters. You can define the target repository (or repositories) using AQL. By default, all available repositories are searched when no target repository is specified in AQL. Just like searching with the GET HTTP method, these default repository names are cached.

##### 2.45.4.4.1 HTTP Request method and query parameters

Many URI parameters are not supported because their functions are covered in AQL. The following table shows more details:

Parameter Type	Parameter	Support	Description
Common URI parameters	include-total	Yes	When only one repository is being searched the field, <i>total</i> is returned. When more than one repository is being searched, a Map named <i>total-for-each</i> is returned rather than <i>total</i> .
	inline	Yes	
	page	No	An exception is thrown when this parameter is specified.

Parameter Type	Parameter	Support	Description
	items-per-page	No	An exception is thrown when this parameter is specified.
	start	Yes	Defines the starting index of the search range for each repository. The default value is 1.
	limit	Yes	Defines the maximum size of the search range for each repository. The default value is 100.
	sort	No	
	view	No	
Search URI parameters	q	No	
	locations	No	
	collections	No	
	object-type	No	
	facet	No	
	facet-value-constraints	No	
	facet-id-constraints	Yes	<p>It contains pairs of facet id and its constraints.</p> <p>For example, the expression below has two facet constraints:</p> <ul style="list-style-type: none"> <li>• id1 is the facet for the <code>r_object_type</code> property</li> <li>• id2 is the facet for the <code>owner_name</code> property</li> </ul> <p><code>facet-id-constraints=id1%3Ddm_document,id2%3dAdministrator</code></p>
	query-locale	Yes	<p>A locale used to evaluate the query and parse words into a language dependent token.</p> <p>Only supported with <i>xPlore</i>.</p> <p>The default value is <i>en</i>.</p>

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.45.4.4.2 Request media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.45.4.4.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.45.4.4.4 Request body

For more information, see [Appendix E, AQL on page 1059](#) for a complete definition.

When working with this resource, please note the following items:

- The *repositories* element specified in the *AQL* defines the scope of the searching procedure.
- Repository names are validated before the search begins. An error message is shown when an invalid repository name is specified.
- When the *repositories* element is not provided, then by default all repositories are searched.

#### ➤ Example 2-181: An XML request

```
<search xmlns="http://identifiers.emc.com/vocab/documentum"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <types>
        <type>dm_document</type>
    </types>
    <repositories>
        <repository>REPO8</repository>
        <repository>REPO1</repository>
    </repositories>
    <expression-set operator="AND">
        <expressions>
            <fulltext>REST</fulltext>
            <property name="object_name" operator="BEGINS_WITH">Doc</property>
            <property-list name="owner_name" operator="IN">
                <values>
                    <value>Administrator</value>
                </values>
            </property-list>
            <property-range name="r_full_content_size" operator="BETWEEN">
                <from>130000</from>
                <to>900000</to>
            </property-range>
        </expressions>
    </expression-set>
</search>
```

```
</expression-set>
</search>
```



#### ▶ Example 2-182: A JSON request

```
{
  "types": [ "dm_document" ],
  "repositories": [ "REP01" ],
  "expression-set": {
    "expression-type": "expression-set",
    "operator" : "AND",
    "expressions": [
      {
        "expression-type": "property",
        "name": "object_name",
        "operator": "CONTAINS",
        "value": "Doc"
      },
      {
        "expression-type": "fulltext",
        "value": "rest"
      },
      {
        "expression-type": "property-range",
        "name": "r_content_size",
        "operator": "BETWEEN",
        "from": "0",
        "to": "100000"
      }
    ]
  }
}
```



##### 2.45.4.4.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.45.4.4.6 Response media types

N/A

#### 2.45.4.4.7 Response status

- 200 - Retrieved successfully
- 201, 204, 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.45.4.4.8 Response body

The Response body is the same as the one shown for the GET method above.

##### ➤ Example 2-183: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/search</id>
  <title>Search results</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2018-12-07T06:26:06.665+00:00</updated>
  <dm:total-for-each xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    <dm:REPO>0</dm:REPO>
    <dm:REPO1>2</dm:REPO1>
  </dm:total-for-each>
  <link rel="self" href="http://localhost:8080/dctm-rest/search.xml?start=1
&limit=1"/>
  <link rel="search" hreftemplate="http://localhost:8080/dctm-rest/
search.xml{?collections,facet,include-total,inline,limit,locations,
object-type,q,start,timezone,view}"/>
  <entry>
    <id>0c00000280002104</id>
    <title>auditadmin1</title>
    <author>
      <name>dmadmin1</name>
    </author>
    <updated>2018-06-13T03:20:15.000+00:00</updated>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO1/objects/
0c00000280002104.xml"/>
    <content type="application/xml"
             src="http://localhost:8080/dctm-rest/repositories/REPO1/objects/
0c00000280002104.xml"/>
    <relevance:score xmlns:relevance="http://a9.com/-/opensearch/extensions/
relevance/1.0/">1.0</relevance:score>
    <dm:source xmlns:dm="http://identifiers.emc.com/vocab/documentum">REPO1
  </dm:source>
  </entry>
</feed>
```



 Example 2-184: JSON response

```
{  
    "id": "http://localhost:8080/dctm-rest/search",  
    "title": "Search results",  
    "author": [  
        {  
            "name": "OpenText Documentum"  
        }  
    ],  
    "updated": "2018-12-07T06:27:02.819+00:00",  
    "total-for-each": {  
        "REPO": 0,  
        "REPO1": 2  
    },  
    "links": [  
        {  
            "rel": "self",  
            "href": "http://localhost:8080/dctm-rest/search?start=1&limit=1"  
        },  
        {  
            "rel": "search",  
            "hreftemplate": "http://localhost:8080/dctm-rest/search{?collections,  
                facet,include-total,inline,limit,locations,  
                object-type,q,start,timezone,view}"  
        }  
    ],  
    "entries": [  
        {  
            "id": "0c00000280002104",  
            "title": "auditadmin1",  
            "author": [  
                {  
                    "name": "dmadmin1"  
                }  
            ],  
            "updated": "2018-06-13T03:20:15.000+00:00",  
            "links": [  
                {  
                    "rel": "edit",  
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO1/  
                        objects/0c00000280002104"  
                }  
            ],  
            "content": {  
                "type": "application/vnd.emc.documentum+json",  
                "src": "http://localhost:8080/dctm-rest/repositories/REPO1/objects  
                    /0c00000280002104"  
            },  
            "score": "1.0",  
            "source": "REPO1"  
        }  
    ]  
}
```



## 2.45.5 Authentication

This section describes the authentication modes used for this resource and some important details.

### 2.45.5.1 Credentials usages

Every Request for this resource, which uses a URL pattern like `/search*`, must contain login credentials for authentication.

The login credentials provided by users have two purposes:

- Retrieve a global OpenText™ Documentum™ Content Management Foundation Java API session from the Global Registry Repository of Documentum CM Server to use for all subsequent search operations.
- Provide authentication information for every target repository.

#### 2.45.5.1.1 Search authentication

- Every repository including the global registry repository must have a user that has the same username. Further, when the underlying authentication scheme requests a password, then this user must also have the same password.
- When the credentials provided are only valid for just one repository (or for just part of all of the specified repositories), then the REST server will throw an authentication error and no results will be returned.

### 2.45.5.2 Authentication modes

The following modes can be used to authenticate for this resource:

- `Http Basic('basic')`
- `Client Token('ct-basic')`
- OTDS
  - `otds_password`
  - `otds_token`
  - `ct-otds_password`
  - `ct-otds_token`
  - `otds_token-basic`
  - `ct-otds_token-basic`

#### 2.45.5.2.1 Configuration for client token

Any authentication mode that has a *ct-* prefix uses REST CT (Client Token) and a Documentum CM Server login ticket for authentication. When one of these authentication modes are used to search repositories, every repository being searched must be configured to trust other repositories, at minimum they must be configured to trust the global registry repository. Also, all repositories being searched must allow the use of a global login ticket, which must have the same Login Ticket Key (LTK), from another repository in order to insure that the global login ticket works.

For more information, see *OpenText Documentum Content Management - Server Fundamentals Guide* (EDCCS250400-GGD).

#### 2.45.5.2.2 Configuration for OTDS

A repository is referred to as a *Resource* in OTDS.

A global user must exist, or be created, in every repository before that global user can be used with an OTDS related authentication mode to login. The global user must be able to access every repository in OTDS. Therefore, an SSO (Single Sign On) access token, which can be used for a Multiple Repository Search and other single repository resources, is issued to the global user.

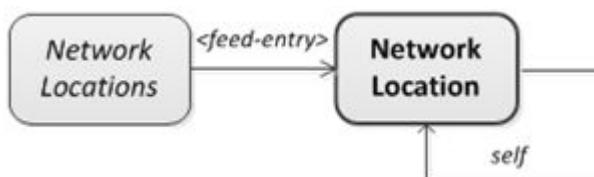
## 2.46 Network Location (s)

### 2.46.1 Network Location

The Network Location resource represents a network location, which is an instance of `dm_network_location_map`, that is stored in a repository.

#### 2.46.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Network Locations” on page 449 and About the Diagram on page 13.

### 2.46.1.2 Feed

No feed support.

### 2.46.1.3 Link relations

Link relation	Description	Resource reference
self	This network location resource.	<a href="#">"Network Location" on page 446</a>

### 2.46.1.4 Operations

#### 2.46.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the metadata of a network location (an instance of <code>dm_network_location_map</code> ) in the repository.

#### 2.46.1.4.2 Get a Network Location

Retrieves the metadata of a network location (an instance of `dm_network_location_map`) specified by the identifier.

##### 2.46.1.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- `view`
- `links`

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.46.1.4.2.2 Request media types

N/A

#### 2.46.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.46.1.4.2.4 Request body

N/A

#### 2.46.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.46.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.46.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.46.1.4.2.8 Response body

XML or JSON representation of the network location. The response contains properties of a network location, depending on the setting of the `view` query parameter. By default, all properties are returned.



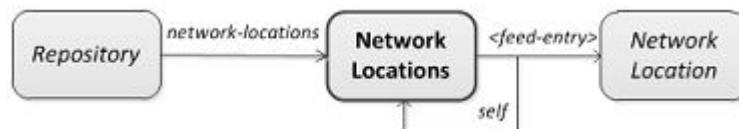
**Note:** The network location in the URL is encoded.

## 2.46.2 Network Locations

The Network Locations collection resource represents the collection of network locations (instances of `dm_network_location_map`) in a repository.

### 2.46.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.46.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of network locations
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Network Location” on page 446
- Entry ID: Network location URI
- Entry Title: Display name of the network location.
- Entry Updated: `r_modify_date` of the network location (the `dm_network_location_map` instance).

### 2.46.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of network locations	“Network Locations” on page 449
first, last, next, previous	Pagination links.	“Network Locations” on page 449

## 2.46.2.4 Operations

### 2.46.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the metadata of a collection of network locations (instances of <code>dm_network_location_map</code> ) in the repository.

### 2.46.2.4.2 Get Network Locations

Retrieves the metadata of a collection of network locations (instances of `dm_network_location_map`) in the repository.

#### 2.46.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- `sort`
- `view`
- `inline`
- `page`
- `items-per-page`
- `include-total`
- `filter`
- `links`

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.46.2.4.2.2 Request media types

N/A

#### 2.46.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.46.2.4.2.4 Request body

N/A

#### 2.46.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.46.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.46.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.46.2.4.2.8 Response body

XML or JSON representation of the collection of network locations.

- The body contains a list of network locations.
- Each object may contain all properties of the network location, depending on the setting of the query parameter `view`.
- The returned network locations collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order by `netloc_ident`.
- The network location in the URL is encoded.

## 2.47 Object Aspect (s)

### 2.47.1 Object Aspect

The Object Aspect resource is only used for detaching an aspect from a specified object.

#### 2.47.1.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “Object Aspects” on page 454 and About the Diagram on page 13.

#### 2.47.1.2 Link relations

None.

#### 2.47.1.3 Operations

##### 2.47.1.3.1 Supported HTTP methods

Method	Description
DELETE	Detach an aspect from an object

##### 2.47.1.3.2 Detach an object aspect

Detach an aspect from an object.

###### 2.47.1.3.2.1 HTTP Request method and query parameters

DELETE

#### 2.47.1.3.2.2 Server Accepted Request media types

None.

#### 2.47.1.3.2.3 Request headers

- Authorization

For more details about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.47.1.3.2.4 Request body

None.

#### ➡ Example 2-185: Detaching an Aspect

```
DELETE http://localhost:8080/dctm-rest/repositories/REPO/objects/  
090000580002523/aspects/my_aspect_1
```



#### 2.47.1.3.2.5 Response headers

None.

#### 2.47.1.3.2.6 Supported Response media types

None.

#### 2.47.1.3.2.7 Response status

- 204 - Detached successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.47.1.3.2.8 Response body

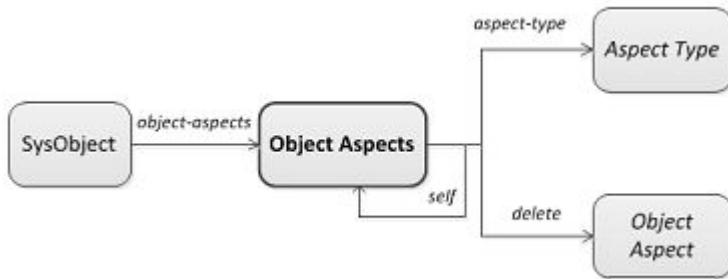
HTTP 204 No Content status upon a successful detach operation. The response body contains no content.

## 2.47.2 Object Aspects

The Object Aspects resource represents the aspects attached to a specified object in a repository. Using this resource, you can attach extra properties to certain objects.

### 2.47.2.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “[SysObject](#)” on page 692, “[Document](#)” on page 214, “[Folder](#)” on page 233, “[Cabinet](#)” on page 132, and [About the Diagram](#) on page 13.

### 2.47.2.2 Feed

No feed support.

### 2.47.2.3 Link relations

The following table lists general link relations in the Object Aspects resource.

Link relation	Description	Resource reference
self	This Object Aspects resource	<a href="#">“Object Aspects” on page 454</a>
aspect-type [1]	Type definition of an aspect	<a href="#">“Aspect Type” on page 73</a>
delete [1]	Detach an aspect from the SysObject	<a href="#">“Object Aspect” on page 452</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
<http://identifiers.emc.com/linkrel/>.

## 2.47.2.4 Operations

### 2.47.2.4.1 Supported HTTP methods

Method	Description
GET	GET all aspects attached to a SysObject
POST	Attach one or more aspects to a SysObject

### 2.47.2.4.2 Get Attached Aspects

GET all aspects attached to a specified SysObject.

#### 2.47.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameter:

- links

For more details about the query parameters, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.47.2.4.2.2 Server accepted request media types

None.

#### 2.47.2.4.2.3 Request headers

- Accept
- Authorization

For more details about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.47.2.4.2.4 Request body

None.

#### 2.47.2.4.2.5 Response headers

- Content-Type

For more details about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.47.2.4.2.6 Supported response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing)

#### 2.47.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.47.2.4.2.8 Response body

XML or JSON representation of the aspects attached to the `SysObject`. The response body mainly contains two parts:

- The first part is a set of aspect elements, each of which represents an aspect attached to the `SysObject`.
- The second part is the `links` block. Each attached aspect has two related links:
  - An aspect-type link pointing to the corresponding aspect type definition. For more information, see [Aspect Type](#).
  - A delete link enabling you to detach the aspect. For more information, see [Object Aspect](#).

##### *XML representation*

```
<?xml version="1.0" encoding="UTF-8"?>
<object-aspects xmlns="http://identifiers.emc.com/vocab/documentum"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <aspect>my_aspect_1</aspect>
  <aspect>dm_checkin_control</aspect>
  <aspect>dm_version_behavior</aspect>
  <links>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000580002523/aspects" rel="self"/>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000580002523/aspects/my_aspect_1"
      rel="http://identifiers.emc.com/linkrel/delete"
```

```

    title="my_aspect_1"/>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/aspect-
types/my_aspect_1"
            rel="http://identifiers.emc.com/linkrel/aspect-type"
    title="my_aspect_1"/>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/
            aspects/dm_checkin_control"
            rel="http://identifiers.emc.com/linkrel/delete"
    title="dm_checkin_control"/>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/aspect-
types/dm_checkin_control"
            rel="http://identifiers.emc.com/linkrel/aspect-type"
    title="dm_checkin_control"/>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/
            aspects/dm_version_behavior"
            rel="http://identifiers.emc.com/linkrel/delete"
    title="dm_version_behavior"/>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/aspect-
types
            /dm_version_behavior"
            rel="http://identifiers.emc.com/linkrel/aspect-type"
    title="dm_version_behavior"/>
    </links>
</object-aspects>
```

*JSON representation*

```
{
    "aspects" : [
        "my_aspect_1",
        "dm_checkin_control",
        "dm_version_behavior"
    ],
    "links" : [
        {
            "rel" : "self",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/aspects"
        },
        {
            "rel" : "http://identifiers.emc.com/linkrel/delete",
            "title" : "my_aspect_1",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/
            aspects/my_aspect_1"
        },
        {
            "rel" : "http://identifiers.emc.com/linkrel/aspect-type",
            "title" : "my_aspect_1",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/aspect-types/
my_aspect_1"
        },
        {
            "rel" : "http://identifiers.emc.com/linkrel/delete",
            "title" : "dm_checkin_control",
            "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
```

```
0900000580002523/
    aspects/ dm_checkin_control"
},
{
    "rel" : "http://identifiers.emc.com/linkrel/aspect-type",
    "title" : "dm_checkin_control",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/aspect-types/
dm_checkin_control"
},
{
    "rel" : "http://identifiers.emc.com/linkrel/delete",
    "title" : "dm_version_behavior",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/
    aspects/dm_version_behavior"
},
{
    "rel" : "http://identifiers.emc.com/linkrel/aspect-type",
    "title" : "dm_version_behavior",
    "href" : "http://localhost:8080/dctm-rest/repositories/REPO/aspect-types/
dm_version_behavior"
}
]
```

#### 2.47.2.4.3 Attach aspects to a SysObject

Attach one or more aspects to a specified SysObject.

##### 2.47.2.4.3.1 HTTP Request method and query parameters

POST

##### 2.47.2.4.3.2 Server accepted request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.47.2.4.3.3 Request headers

- Authorization
- Content-Type
- Accept

For more details about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.47.2.4.3.4 Request body

To attach aspects to a `SysObject`, specify the aspect type names in the request body.

*XML representation*

```
POST http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/aspects
Accept: application/vnd.emc.documentum+xml
Content-Type: application/vnd.emc.documentum+xml
<object-aspects xmlns="http://identifiers.emc.com/vocab/documentum"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <aspect>my_aspect_1</aspect>
    <aspect>dm_checkin_control</aspect>
    <aspect>dm_version_behavior</aspect>
</object-aspects>
```

*JSON representation*

```
POST http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002523/aspects
Accept: application/vnd.emc.documentum+json
Content-Type: application/vnd.emc.documentum+json{
    "aspects": [
        "my_aspect_1", "dm_checkin_control", "dm_version_behavior"
    ]
}
```

#### 2.47.2.4.3.5 Response headers

- Content-Type

For more details about HTTP headers, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.47.2.4.3.6 Supported response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xmlPalermo
- application/json

#### 2.47.2.4.3.7 Response status

- 201 - Attached successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.47.2.4.3.8 Response body

XML or JSON representation of all aspects attached to the SysObject.

#### 2.47.2.5 Add aspect properties to object properties

You can add aspect properties to an object's properties. After the updated object will contain the aspect's properties in its representation. The property names attached by using aspects are displayed according to the following pattern:

`<aspect_name>. <property_name>`

You can update the aspect properties in the same way as other common object properties.

➡ **Example 2-186: Updated object properties with aspect properties**

```
{
  "properties" : {
    "r_object_id" : "0900000580003c83",
    "object_name" : "my doc",
    "r_object_type" : "dm_document",
    "my_aspect_1.annotation_name" : "anno",
    "my_aspect_1.module_name" : "mod",
    "my_aspect_1.annotation_metadata" : "meta",
    "my_aspect_1.i_partition" : 0,
  }, "links" : [...]
}
```

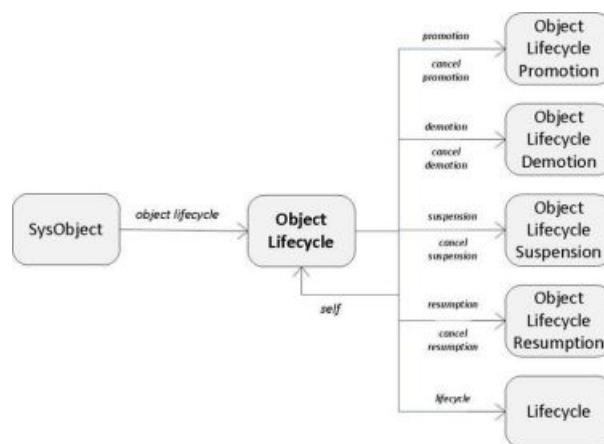


## 2.48 Object Lifecycle

The current information of the attached Lifecycle object.

### 2.48.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagrams](#) on page 13.

## 2.48.2 Feed

No feed support.

## 2.48.3 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">“Object Lifecycle” on page 460</a>
promotion [1]	Promote the object Lifecycle to the next state	<a href="#">“Object Lifecycle Promotion” on page 475</a>
cancel-promotion [1]	Cancel a scheduled promotion	<a href="#">“Object Lifecycle Promotion” on page 475</a>
demotion [1]	Demote the object Lifecycle to the previous state	<a href="#">“Object Lifecycle Demotion” on page 469</a>
cancel-demotion [1]	Cancel a scheduled demotion	<a href="#">“Object Lifecycle Demotion” on page 469</a>
suspension [1]	Suspend the object Lifecycle to the exceptional state	<a href="#">“Object Lifecycle Suspension” on page 487</a>
cancel-suspension [1]	Cancel a scheduled suspension	<a href="#">“Object Lifecycle Suspension” on page 487</a>
resumption [1]	Resume the object Lifecycle to the normal state	<a href="#">“Object Lifecycle Resumption” on page 481</a>
cancel-resumption [1]	Cancel a scheduled resumption	<a href="#">“Object Lifecycle Resumption” on page 481</a>
Lifecycle [1]	The attached Lifecycle	<a href="#">“Lifecycle” on page 370</a>
parent	The object resource	<a href="#">“SysObject” on page 692</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.48.4 Operations

#### 2.48.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the Lifecycle status of the object that is currently attached.
PUT	Attach a Lifecycle to the object.
DELETE	Detach a Lifecycle from the object that is currently attached.

#### 2.48.4.2 Get an Object Lifecycle

Retrieves the status of an object's currently attached lifecycle.

##### 2.48.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.48.4.2.2 Request media types

N/A

##### 2.48.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.48.4.2.4 Request body

N/A

##### 2.48.4.2.5 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.48.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.48.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.48.4.2.8 Response body

##### Example 2-187: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<object-lifecycle
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <lifecycle-id>4600000180001165</lifecycle-id>
    <lifecycle-name>lifecycle-for-doc</lifecycle-name>
    <current-state>review</current-state>
    <previous-state>draft</previous-state>
    <next-state>publish</next-state>
    <able-to-promote>true</able-to-promote>
    <able-to-demote>true</able-to-demote>
    <able-to-suspend>true</able-to-suspend>
    <able-to-resume>false</able-to-resume>
    <schedules>
        <schedule action="promote" from="review"
to="publish">2020-01-01T12:00:00.000+00:00</schedule>
    </schedules>
    <links>
        <link rel="self"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
            lifecycle.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/lifecycle"
            href="http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
            4600000180001165.xml"/>
        <link rel="parent"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/promotion"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/
            lifecycle/promotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/demotion"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/
            lifecycle/demotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/suspension"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/
            lifecycle/suspension.xml"/>
    </links>
</object-lifecycle>
```

```

0900000018000117c/
    lifecycle/suspension.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/cancel-promotion"
        title="2020-01-01T12:00:00.000+00:00"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000018000117c/
    lifecycle/promotion.xml?
schedule=2020-01-01T12:00:00.000%2B00:00"/>
</links>
</object-lifecycle>
```



#### ➤ Example 2-188: JSON response

```
{
    "lifecycle-id": "4600000180001165",
    "lifecycle-name": "lifecycle-for-doc",
    "current-state": "review",
    "previous-state": "draft",
    "next-state": "publish",
    "able-to-promote": true,
    "able-to-demote": true,
    "able-to-suspend": true,
    "able-to-resume": false,
    "schedules": [
        {
            "action": "promote",
            "schedule": "2020-01-01T12:00:00.000+00:00",
            "from": "review",
            "to": "publish"
        }
    ],
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000018000117c/
    lifecycle"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/lifecycle",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000180001165"
        },
        {
            "rel": "parent",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000018000117c"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/promotion",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000018000117c/
    lifecycle/promotion"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/demotion",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000018000117c/
    lifecycle/demotion"
        }
    ]
}
```

```

    },
    {
      "rel": "http://identifiers.emc.com/linkrel/suspension",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle/suspension"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/cancel-promotion",
      "title": "2020-01-01T12:00:00.000+00:00",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle/promotion?schedule=2020-01-01T12:00:00.000%2B00:00"
    }
  ]
}

```



### 2.48.4.3 Attach a lifecycle

Attaches a Lifecycle to the current object.

#### 2.48.4.3.1 HTTP Request method and query parameters

PUT

#### 2.48.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.48.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.48.4.3.4 Request body

##### Example 2-189: XML request

```

<object-lifecycle>
  <lifecycle-id>XXX</lifecycle-id>
  <current-state>XXX</current-state>
  <aliaset-name>XXX</aliaset-name>
</object-lifecycle>

```



 **Example 2-190: JSON request**

```
{  
    "lifecycle-id" : "XXX",  
    "current-state" : "XXX",  
    "aliaset-name" : "XXX"  
}
```

**Query parameters**

- The `lifecycle-id` property specifies the id of the Lifecycle to be attached the object. When the type has a default Lifecycle defined, this property is optional.
- The `current-state` specifies the name of a normal state in which the object is attached to a Lifecycle. When you want to attach an object to a specified state, the `current-state` parameter must have its `allow-attach` property set to `true`, then you can define the state name using the `current-state` parameter.
- The `current-state` parameter is optional. However, when a value for the `current-state` parameter is not provided, the base state is attached by default.
- The `aliaset-name` parameter specifies the name of an alias set. The `aliaset-name` parameter must contain a valid alias name, which must be defined in the Lifecycle.



**Note:** The request body is optional and empty when none of the above three parameters are defined.

**2.48.4.3.5 Response headers****Content-Type**

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.48.4.3.6 Response media types**

- `application/vnd.emc.documentum+xml`
- `application/xml`
- `application/vnd.emc.documentum+json`
- `application/json`

#### 2.48.4.3.7 Response status

- 201 - Object Lifecycle successfully attached.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.48.4.3.8 Response body

The response body is the same as the one shown in the [GET an Object Lifecycle samples](#).

##### ➡ Example 2-191: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<object-lifecycle
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <lifecycle-id>4600000180001165</lifecycle-id>
    <lifecycle-name>lifecycle-for-doc</lifecycle-name>
    <current-state>review</current-state>
    <previous-state>draft</previous-state>
    <next-state>publish</next-state>
    <able-to-promote>true</able-to-promote>
    <able-to-demote>true</able-to-demote>
    <able-to-suspend>true</able-to-suspend>
    <able-to-resume>false</able-to-resume>
    <links>xxx</links>
</object-lifecycle>
```



##### ➡ Example 2-192: JSON response

```
{
    "lifecycle-id": "4600000180001165",
    "lifecycle-name": "lifecycle-for-doc",
    "current-state": "review",
    "previous-state": "draft",
    "next-state": "publish",
    "able-to-promote": true,
    "able-to-demote": true,
    "able-to-suspend": true,
    "able-to-resume": false,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/090000018000117c/lifecycle"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/lifecycle",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/4600000180001165"
        },
        {
            "rel": "parent",
        }
    ]
}
```

```
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/090000018000117c",
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/promotion",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/090000018000117c/lifecycle/promotion"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/demotion",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/090000018000117c/lifecycle/demotion"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/suspension",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/090000018000117c/lifecycle/suspension"
    }
]
```



#### 2.48.4.4 Detach a Lifecycle

Detach the Lifecycle from a specified object. When a Lifecycle has been detached from an object, any existing or scheduled Lifecycle operations for that object are automatically cancelled.

##### 2.48.4.4.1 HTTP Request method and query parameters

DELETE

##### 2.48.4.4.2 Request media types

N/A

##### 2.48.4.4.3 Request headers

Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.48.4.4.4 Request body**

N/A

**2.48.4.4.5 Response headers**

Content-Type, etc.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.48.4.4.6 Response media types**

N/A

**2.48.4.4.7 Response status**

- 204 - Object Lifecycle successfully detached.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.48.4.4.8 Response body**

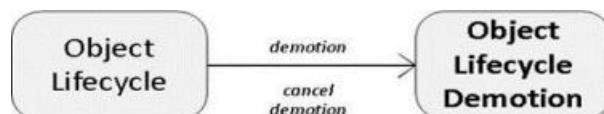
N/A

## 2.49 Object Lifecycle Demotion

This resource allows the demotion of an Object Lifecycle.

### 2.49.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Object Lifecycle” on page 460 and [About the Diagrams on page 13](#).

## 2.49.2 Feed

No feed support.

## 2.49.3 Link relations

Link relation	Description	Resource reference
self	This resource	"Object Lifecycle Demotion" on page 469
object-lifecycle [1]	The object Lifecycle resource	"Object Lifecycle" on page 460
parent	The object resource	"SysObject" on page 692
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

## 2.49.4 Operations

### 2.49.4.1 Supported HTTP methods

Method	Description
PUT	Move the object from a normal state back to the previous normal state or back to its base state.
DELETE	Cancel the scheduled demotion.

### 2.49.4.2 Demote a lifecycle

Demotes a Lifecycle to its previous state, or to its base state.

#### 2.49.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	to-base	Whether to demote the object to the base state.  This parameter is ignored when a value for the schedule parameter has been provided. [1]	string	true, false	false

HTTP method	Query parameters	Description	Data type	Value range	Default value
	schedule	When this parameter is provided, the demotion is scheduled for the time that is set. [2]	datetime	Any valid datetime	null

[1] A scheduled demotion can only be made to the previous normal state. In this case, when the schedule parameter has been set, the to-base parameter is ignored.

[2] To schedule a demotion, your user account must allow this action in its user privilege and object permission constraints.



### Caution

Any manual demote operation, where `schedule=null`, causes the cancellation of all existing scheduled operations, regardless of whether the operation succeeds or fails.

#### 2.49.4.2.2 Request media types

N/A

#### 2.49.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.49.4.2.4 Request body

N/A

#### 2.49.4.2.5 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.49.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.49.4.2.7 Response status

- 200 - Demote or scheduled demotion was successful
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.49.4.2.8 Response body

##### Example 2-193: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<object-lifecycle xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <lifecycle-id>4600000180001165</lifecycle-id>
    <lifecycle-name>lifecycle-for-doc</lifecycle-name>
    <current-state>review</current-state>
    <previous-state>draft</previous-state>
    <next-state>publish</next-state>
    <able-to-promote>true</able-to-promote>
    <able-to-demote>true</able-to-demote>
    <able-to-suspend>true</able-to-suspend>
    <able-to-resume>false</able-to-resume>
    <schedules>
        <schedule action="demote" from="review" to="draft">
            2020-01-01T12:00:00.000+00:00
        </schedule>
    </schedules>
    <links>
        <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/objects/
            090000018000117c/lifecycle.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/lifecycle"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
            4600000180001165.xml"/>
        <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/
REPO/
            objects/090000018000117c.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/demotion"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle/demotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/demotion"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle/demotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/suspension"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle/suspension.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/cancel-demotion"
```

```

        title="2020-01-01T12:00:00.000+00:00"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
          090000018000117c/lifecycle/demotion.xml?
        schedule=2020-01-01T12:00:00.000%2B00:00"/>
    </links>
</object-lifecycle>
```



#### Example 2-194: JSON response

```
{
  "lifecycle-id": "4600000180001165",
  "lifecycle-name": "lifecycle-for-doc",
  "current-state": "review",
  "previous-state": "draft",
  "next-state": "publish",
  "able-to-promote": true,
  "able-to-demote": true,
  "able-to-suspend": true,
  "able-to-resume": false,
  "schedules": [
    {
      "action": "demote",
      "schedule": "2020-01-01T12:00:00.000+00:00",
      "from": "review",
      "to": "draft"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        090000018000117c/
        lifecycle"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/lifecycle",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
        4600000180001165"
    },
    {
      "rel": "parent",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        090000018000117c"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/demotion",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        090000018000117c/
        lifecycle/demotion"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/demotion",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        090000018000117c/
        lifecycle/demotion"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/suspension",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        090000018000117c/
        lifecycle/demotion"
    }
  ]
}
```

```

090000018000117c/
    lifecycle/suspension"
},
{
    "rel": "http://identifiers.emc.com/linkrel/cancel-demotion",
    "title": "2020-01-01T12:00:00.000+00:00",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
    lifecycle/demotion?schedule=2020-01-01T12:00:00.000%2B00:00"
}
]
}

```



### 2.49.4.3 Cancel a scheduled demotion

Cancel a scheduled demotion for the object.

#### 2.49.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	schedule	This is a required parameter. The execution time of the scheduled demotion. [1]	datetime	Any valid datetime	null

[1] To cancel a scheduled demotion, your user account must allow this action in its user privilege constraints.

#### 2.49.4.3.2 Request media types

N/A

#### 2.49.4.3.3 Request headers

- Authorization
- Accept

For more information, see Appendix D, REST Common Definition - HTTP Headers on page 1055.

**2.49.4.3.4 Request body**

N/A

**2.49.4.3.5 Response headers**

N/A

**2.49.4.3.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.49.4.3.7 Response status**

- 204 - Scheduled demotion successfully canceled.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.49.4.3.8 Response body**

N/A

## 2.50 Object Lifecycle Promotion

This resource allows for the promotion of an Object Lifecycle.

### 2.50.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Object Lifecycle” on page 460 and [About the Diagrams on page 13](#).

## 2.50.2 Feed

No feed support.

## 2.50.3 Link relations

Link relation	Description	Resource reference
self	This resource	"Object Lifecycle Promotion" on page 475
object-lifecycle [1]	The object Lifecycle resource	"Object Lifecycle" on page 460
parent	The object resource	"SysObject" on page 692
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/.</code>		

## 2.50.4 Operations

### 2.50.4.1 Supported HTTP methods

Method	Description
PUT	Move the object from one normal state to the next normal state.
DELETE	Cancel the scheduled promotion.

### 2.50.4.2 Promote a lifecycle

Promotes a Lifecycle to its next normal state.

#### 2.50.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	force	Force the object to be promoted, regardless of any entry criteria for the target state.	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	trial	Test whether the promotion can be done. This does not promote the object. [1]	string	<i>true, false</i>	<i>false</i>
	schedule	When provided, this parameter schedules the promotion for the date and time set as its value. [2]	datetime	Any valid datetime	null

[1] Providing a value for both the `schedule` and `trial` parameters causes a 400 error.

[2] To schedule a promotion, your user account must allow this action in its user privilege and object permission constraints.



### Caution

Any manual promote operation, where `schedule=null` and `trial=false`, causes the cancellation of all existing scheduled operations, regardless of whether the operation succeeds or fails.

#### 2.50.4.2.2 Request media types

N/A

#### 2.50.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.50.4.2.4 Request body**

N/A

**2.50.4.2.5 Response headers**

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.50.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.50.4.2.7 Response status**

- 200 - Promote or scheduled promotion was successful
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.50.4.2.8 Response body****Example 2-195: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<object-lifecycle xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <lifecycle-id>4600000180001165</lifecycle-id>
    <lifecycle-name>lifecycle-for-doc</lifecycle-name>
    <current-state>review</current-state>
    <previous-state>draft</previous-state>
    <next-state>publish</next-state>
    <able-to-promote>true</able-to-promote>
    <able-to-demote>true</able-to-demote>
    <able-to-suspend>true</able-to-suspend>
    <able-to-resume>false</able-to-resume>
    <schedules>
        <schedule action="promote" from="review" to="publish">
            2020-01-01T12:00:00.000+00:00
        </schedule>
    </schedules>
    <links>
        <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/objects/
090000018000117c/lifecycle.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/lifecycle"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000180001165.xml"/>
```

```

        <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/
REPO/
                                         objects/090000018000117c.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/promotion"
              href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/promotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/demotion"
              href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/demotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/suspension"
              href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/suspension.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/cancel-promotion"
              title="2020-01-01T12:00:00.000+00:00"
              href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/promotion.xml?
schedule=2020-01-01T12:00:00.000%2B00:00"/>
    </links>
</object-lifecycle>
```



#### Example 2-196: JSON response

```
{
  "lifecycle-id": "4600000180001165",
  "lifecycle-name": "lifecycle-for-doc",
  "current-state": "review",
  "previous-state": "draft",
  "next-state": "publish",
  "able-to-promote": true,
  "able-to-demote": true,
  "able-to-suspend": true,
  "able-to-resume": false,
  "schedules": [
    {
      "action": "promote",
      "schedule": "2020-01-01T12:00:00.000+00:00",
      "from": "review",
      "to": "publish"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
                                         lifecycle"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/lifecycle",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000180001165"
    },
    {
      "rel": "parent",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/promotion",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/promotion.xml"
    }
  ]
}
```

```

        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
            lifecycle/promotion"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/demotion",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
            lifecycle/demotion"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/suspension",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
            lifecycle/suspension"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/cancel-promotion",
        "title": "2020-01-01T12:00:00.000+00:00",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
            lifecycle/promotion?schedule=2020-01-01T12:00:00.000%2B00:00"
    }
]
}

```



#### 2.50.4.3 Cancel a scheduled promotion

Cancel a scheduled promotion for the object.

##### 2.50.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	schedule	This is a required parameter. The execution time of the scheduled promotion [1]	datetime	Any valid datetime	null

[1] To cancel a scheduled promotion, your user account must allow this action in its user privilege constraints.

**2.50.4.3.2 Request media types**

N/A

**2.50.4.3.3 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.50.4.3.4 Request body**

N/A

**2.50.4.3.5 Response headers**

N/A

**2.50.4.3.6 Response media types**

N/A

**2.50.4.3.7 Response status**

- 204 - Scheduled promotion successfully canceled.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.50.4.3.8 Response body**

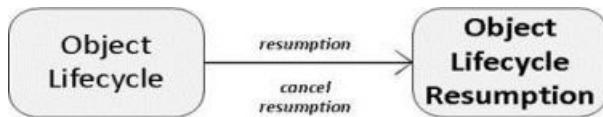
N/A

## 2.51 Object Lifecycle Resumption

This resource allows the resumption of an Object Lifecycle.

### 2.51.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Object Lifecycle” on page 460 and About the Diagrams on page 13.

### 2.51.2 Feed

No feed support.

### 2.51.3 Link relations

Link relation	Description	Resource reference
self	This resumption resource	“Object Lifecycle Resumption” on page 481
object-lifecycle [1]	The object Lifecycle resource	“Object Lifecycle” on page 460
parent	The object resource	“SysObject” on page 692
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.51.4 Operations

#### 2.51.4.1 Supported HTTP methods

Method	Description
PUT	Move the object from an exception state back to the normal state from which it was suspended, or move it back to the base state.
DELETE	Cancel the scheduled resumption.

### 2.51.4.2 Resume a lifecycle

Resumes the lifecycle to its normal state.

#### 2.51.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameter	Description	Data type	Value range	Default value
PUT	to-base	Determines whether to resume the object to its base state.	string	<i>true, false</i>	<i>false</i>
	force	Forces the object to be resumed.	string	<i>true, false</i>	<i>false</i>
	trial	Allows you to test whether the resumption can be done. This will not actually resume the object. [1]	string	<i>true, false</i>	<i>false</i>
	schedule	When provided, this parameter schedules the resumption for the date and time set as its value. [2]	datetime	Any valid datetime	null

[1] Providing a value for both the `schedule` and `trial` parameters causes a 400 error.

[2] To schedule a resumption, your user account must allow this action in its user privilege and object permission constraints.



#### Caution

Any manual resumption operation, where `schedule=null` and `trial=false`, causes the cancellation of all existing scheduled operations, regardless of whether the operation succeeds or fails.

#### 2.51.4.2.2 Request media types

N/A

#### 2.51.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.51.4.2.4 Request body

N/A

#### 2.51.4.2.5 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.51.4.2.6 Response media types

N/A

#### 2.51.4.2.7 Response status

- 200 - Resumption successful or resumption successfully scheduled
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.51.4.2.8 Response body

##### ➡ Example 2-197: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<object-lifecycle xmlns="http://identifiers.emc.com/vocab/documentum"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <lifecycle-id>4600000180001165</lifecycle-id>
    <lifecycle-name>lifecycle-for-doc</lifecycle-name>
    <current-state>failed</current-state>
    <resume-state>review</resume-state>
    <able-to-promote>false</able-to-promote>
    <able-to-demote>false</able-to-demote>
    <able-to-suspend>false</able-to-suspend>
    <able-to-resume>true</able-to-resume>
    <schedules>
        <schedule action="resume" from="review">2020-01-01T12:00:00.000+00:00</
        schedule>
    </schedules>
    <links>
```

```

<link rel="self"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle.xml"/>
<link rel="http://identifiers.emc.com/linkrel/lifecycle"
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            lifecycles/
            4600000180001165.xml"/>
<link rel="parent"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c.xml"/>
<link rel="http://identifiers.emc.com/linkrel/resumption"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle/resumption.xml"/>
<link rel="http://identifiers.emc.com/linkrel/cancel-resumption"
      title="2020-01-01T12:00:00.000+00:00"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle/resumption.xml?
            schedule=2020-01-01T12:00:00.000%2B00:00"/>
</links>
</object-lifecycle>
```



### Example 2-198: JSON response

```
{
  "lifecycle-id": "4600000180001165",
  "lifecycle-name": "lifecycle-for-doc",
  "current-state": "failed",
  "resume-state": "draft",
  "able-to-promote": false,
  "able-to-demote": false,
  "able-to-suspend": false,
  "able-to-resume": true,
  "schedules": [
    {
      "action": "resume",
      "schedule": "2020-01-01T12:00:00.000+00:00",
      "from": "failed"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/
            lifecycle"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/lifecycle",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
            4600000180001165"
    },
    {
      "rel": "parent",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/resumption",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
```

```

        "090000018000117c/lifecycle/resumption"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/cancel-resumption",
        "title": "2020-01-01T12:00:00.000+00:00",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            090000018000117c/lifecycle/resumption?
            schedule=2020-01-01T12:00:00.000%2B00:00"
    }
]
}

```



### 2.51.4.3 Cancel a scheduled resumption

Cancel a scheduled resumption for the lifecycle.

#### 2.51.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameter s	Description	Data type	Value range	Default value
DELETE	schedule	This is a required parameter that defines the execution time of the scheduled resumption [1]	datetime	Any valid datetime	null

[1] Your user account must have sufficient privileges that allow you to cancel a scheduled resumption.

#### 2.51.4.3.2 Request media types

N/A

#### 2.51.4.3.3 Request headers

Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.51.4.3.4 Request body**

N/A

**2.51.4.3.5 Response headers**

Content-Type, etc.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.51.4.3.6 Response media types**

N/A

**2.51.4.3.7 Response status**

- 204 - Scheduled Resumption successfully canceled.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.51.4.3.8 Response body**

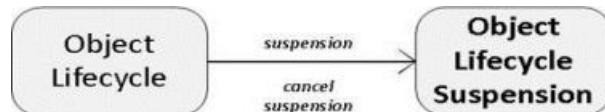
N/A

## 2.52 Object Lifecycle Suspension

This resource allows the Suspension of the object Lifecycle.

### 2.52.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Object Lifecycle” on page 460 and [About the Diagrams on page 13](#).

## 2.52.2 Feed

No feed support.

## 2.52.3 Link relations

Link relation	Description	Resource reference
self	This resumption resource	<a href="#">“Object Lifecycle Suspension” on page 487</a>
object-lifecycle [1]	The object Lifecycle resource	<a href="#">“Object Lifecycle” on page 460</a>
parent	The object resource	<a href="#">“SysObject” on page 692</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
<http://identifiers.emc.com/linkrel/>.

## 2.52.4 Operations

### 2.52.4.1 Supported HTTP methods

Method	Description
PUT	Move the object from the current normal state to the exception state.
DELETE	Cancel the scheduled suspension.

### 2.52.4.2 Suspend a Lifecycle

Suspend a lifecycle to the exception state.

#### 2.52.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	force	Forces the object to be suspended.	string	<i>true, false</i>	<i>false</i>
	trial	Allows you to test whether the suspension can be done. This will not actually suspend the object. [1]	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	schedule	When provided, this parameter schedules the suspension for the date and time set as its value. [2]	datetime	Any valid datetime	null

[1] Providing a value for both the `schedule` and `trial` parameters causes a 400 error.

[2] To schedule a suspension, your user account must allow this action in its user privilege and object permission constraints.



### Caution

Any manual suspension operation, where `schedule=null` and `trial=false`, causes the cancellation of all existing scheduled operations, regardless of whether the operation succeeds or fails.

#### 2.52.4.2.2 Request media types

N/A

#### 2.52.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.52.4.2.4 Request body

N/A

#### 2.52.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.52.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.52.4.2.7 Response status

- 200 - Lifecycle was successfully suspended or scheduled for suspension.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.52.4.2.8 Response body

##### Example 2-199: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<object-lifecycle xmlns="http://identifiers.emc.com/vocab/documentum"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <lifecycle-id>4600000180001165</lifecycle-id>
    <lifecycle-name>lifecycle-for-doc</lifecycle-name>
    <current-state>review</current-state>
    <previous-state>draft</previous-state>
    <next-state>publish</next-state>
    <able-to-promote>true</able-to-promote>
    <able-to-demote>true</able-to-demote>
    <able-to-suspend>true</able-to-suspend>
    <able-to-resume>false</able-to-resume>
    <schedules>
        <schedule action="suspend"
from="review">2020-01-01T12:00:00.000+00:00</schedule>
    </schedules>
    <links>
        <link rel="self"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/lifecycle"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
lifecycles/
4600000180001165.xml"/>
        <link rel="parent"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/promotion"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/promotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/demotion"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/demotion.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/suspension"
            href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/suspension.xml"/>
        <link rel="http://identifiers.emc.com/linkrel/cancel-suspension"
            title="2020-01-01T12:00:00.000+00:00"/>
```

```

        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/lifecycle/suspension.xml?
schedule=2020-01-01T12:00:00.000%2B00:00"/>
</links>
</object-lifecycle>
```



### Example 2-200: JSON response

```
{
  "lifecycle-id": "4600000180001165",
  "lifecycle-name": "lifecycle-for-doc",
  "current-state": "review",
  "previous-state": "draft",
  "next-state": "publish",
  "able-to-promote": true,
  "able-to-demote": true,
  "able-to-suspend": true,
  "able-to-resume": false,
  "schedules": [
    {
      "action": "suspend",
      "schedule": "2020-01-01T12:00:00.000+00:00",
      "from": "review"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/lifecycle",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles/
4600000180001165"
    },
    {
      "rel": "parent",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/promotion",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle/promotion"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/demotion",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle/demotion"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/suspension",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle/suspension"
    }
  ]
}
```

```

    },
    {
      "rel": "http://identifiers.emc.com/linkrel/cancel-suspension",
      "title": "2020-01-01T12:00:00.000+00:00",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000018000117c/
          lifecycle/suspension?schedule=2020-01-01T12:00:00.000%2B00:00"
    }
  }
}

```



### 2.52.4.3 Cancel a Scheduled Suspension

Cancels a scheduled Lifecycle suspension.

#### 2.52.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	schedule	When provided, this parameter defines the execution time of the scheduled suspension that will be canceled [1]	datetime	Any valid datetime	null

[1] To cancel a scheduled suspension, your user account must allow this action in its user privilege and object permission constraints.

#### 2.52.4.3.2 Request media types

N/A

#### 2.52.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.52.4.3.4 Request body**

N/A

**2.52.4.3.5 Response headers**

N/A

**2.52.4.3.6 Response media types**

N/A

**2.52.4.3.7 Response status**

- 204 - Successfully canceled the scheduled Suspension.
- 400, 401, 403, 404, 406, 409, 415, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.52.4.3.8 Response body**

N/A

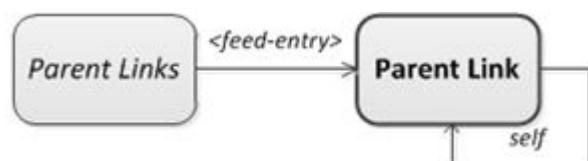
## 2.53 Parent Link (s)

### 2.53.1 Parent Link

The Parent Link resource represents the folder containment relationships between an object and its parent folder.

#### 2.53.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Parent Links” on page 499 and [About the Diagram on page 13](#).

### 2.53.1.2 Feed

No feed support.

### 2.53.1.3 Link relations

Link relation	Description	Resource reference
self	This parent link.	<a href="#">"Parent Link" on page 493</a>

### 2.53.1.4 Operations

#### 2.53.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the folder link between the child object and its parent folder.
PUT	Moves an object from a source folder into a destination folder.
DELETE	Unlinks an object from one of its parent folders.

#### 2.53.1.4.2 Retrieve a folder link

Retrieve the folder link between the child object and its parent folder.

##### 2.53.1.4.2.1 HTTP Request method and query parameters

GET

##### 2.53.1.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.53.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.53.1.4.2.4 Request body

N/A

#### 2.53.1.4.2.5 Response headers

- Content-Type

#### 2.53.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.53.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.53.1.4.2.8 Response body

XML or JSON representation of the folder link between the object and its parent folder. The body contains the URI for the object and the URI for its parent folder. The body also contains the parent ID and child ID.

### 2.53.1.4.3 Move an object from one folder to another

Move an object from a source folder to a destination folder.



#### Notes

- If the object is a folder, this operation also moves the objects under that folder to the destination folder.
- A cabinet object is not allowed to be moved.

#### 2.53.1.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameter	Description	Data type	Value range	Default value
PUT	include-all-versions	Specifies whether or not to move all versions of the object. <ul style="list-style-type: none"><li>• <i>true</i> - Return all versions.</li><li>• <i>false</i> - Only return the current version.</li></ul>	string	<i>true, false</i>	<i>false</i>

#### 2.53.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.53.1.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.53.1.4.3.4 Request body

N/A

#### 2.53.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.53.1.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.53.1.4.3.7 Response status

- 200 - Operation completed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.53.1.4.3.8 Response body

XML or JSON representation of the folder link between the object and the destination folder. The response contains the URI of the new destination folder.

##### Example 2-201: Moving a folder

###### *JSON request*

```
Accept: application/vnd.emc.documentum+json

PUT http://localhost:8080/dctm-rest/repositories/repo/objects/0900000c80000cc4/
    parent-links/0b0004d280002732
{
    "href": "http://localhost:8080/dctm-rest/repositories/repo/folders/
    0b0004d28000ad36"
}
```

###### *JSON response*

```
{
    "child": "http://localhost:8080/dctm-rest/repositories/repo/objects/
    0900000c80000cc4",
    "parent": "http://localhost:8080/dctm-rest/repositories/repo/folders/
    0b0004d28000ad36",
    "links":
    [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
            0900000c80000cc4/
            parent-links/0b0004d28000ad36"
        }
    ]
}
```



#### 2.53.1.4.4 Unlink an object from a parent folder

Unlink an object from one of its parent folders.



##### Notes

- To unlink an object from a primary link, you must have at least the write permission and the change location permission on the object.
- To unlink an object from a secondary link, you must have at least the browse permission and the change location permission on the object.
- If the repository is running under folder security, you must also have at least the write permission on the folder or cabinet from which the object is being unlinked.
- Documents and folders must have at least one link to a folder or cabinet. Therefore, you cannot unlink an object from its only linked folder or cabinet.

##### 2.53.1.4.4.1 HTTP Request method and query parameters

DELETE

##### 2.53.1.4.4.2 Request media types

N/A

##### 2.53.1.4.4.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.53.1.4.4.4 Request body

N/A

##### 2.53.1.4.4.5 Response headers

N/A

#### 2.53.1.4.4.6 Response media types

N/A

#### 2.53.1.4.4.7 Response status

- 204 - Unlinked successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.53.1.4.4.8 Response body

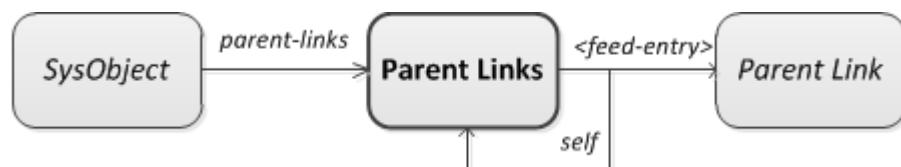
HTTP 204 No Content status upon a successful unlink operation. The response body contains no content.

## 2.53.2 Parent Links

The Parent Links collection resource represents a collection of all folder containment relationships for an object. That is a set of links between an object and its parent folders.

### 2.53.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

### 2.53.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the feed
- Feed Title: Parent links
- Feed Updated: r\_modify\_date of the object
- Supports POST: Yes
- Entry: “[Parent Link](#)” on page 493
- Entry ID: URI of the folder link
- Entry Title: Formatted string with both the child object ID and the parent object ID

- Entry Updated: r\_modify\_date of the child object

### 2.53.2.3 Link relations

Link relation	Description	Resource reference
self	URI for Parent Links collection feed	<a href="#">"Parent Links" on page 499</a>
first, last, next, previous	Pagination links	<a href="#">"Parent Links" on page 499</a>

### 2.53.2.4 Operations

#### 2.53.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of parent links for an object.
POST	Links an object to a new parent folder.

#### 2.53.2.4.2 Get Parent Links

Retrieve all parent links for a given object in a repository.



**Note:** For a cabinet, the parent links collection is empty because a cabinet does not have a parent folder.

##### 2.53.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- page
- items-per-page
- include-total

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

**2.53.2.4.2.2 Request media types**

N/A

**2.53.2.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.53.2.4.2.4 Request body**

N/A

**2.53.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.53.2.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing)

**2.53.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.53.2.4.2.8 Response body**

XML or JSON representation of a collection parent links for the given object.

- The body contains a list of links between the given object and each of its parent folders.
- Each link contains the URI for the given object and the URI for one of its parent folders.
- Each entry contains the parent ID and the child ID.
- Each entry in the collection has a self link referring to the link resource.

- Pagination is supported.

#### 2.53.2.4.3 Link an object to a new folder

Link an object to a new folder or cabinet.

The first execution of link on an object defines the object's primary link, which is the place where the object is stored in the repository. Subsequent executions of link associate the object to other folders or cabinets. These links are called secondary links. This operation only allows you to create a new secondary link for an object. The existing folder links for this object do not change. For more information about how to create primary link during the object creation, see ["Folder Child Objects" on page 282](#).

To create a secondary link, you have at least the browse permission and the change location permission on the object. If the repository is running under folder security, you must also have at least the write permission on the folder or cabinet to which the object is linked.

##### 2.53.2.4.3.1 HTTP Request method and query parameters

POST

##### 2.53.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.53.2.4.3.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.53.2.4.3.4 Request body

XML or JSON representation of a reference to the object's new parent folder.

 **Example 2-202: Example of the request body**

```
{  
    "href":  
        "http://localhost:8080/dctm-rest/repositories/repo/folders/0b0004d280002732"  
}
```



#### 2.53.2.4.3.5 Response headers

- Location
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.53.2.4.3.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.53.2.4.3.7 Response status

- 201 - Link created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.53.2.4.3.8 Response body

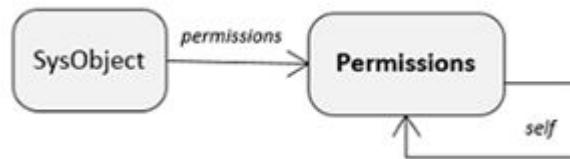
XML or JSON representation of newly-created link between the given object and its new parent folder.

## 2.54 Permissions

The Permissions resource allows a specified user to get basic and extended permissions for a specific accessor on a SysObject. When no accessor is specified, the default accessor is the user that is currently logged into the system.

### 2.54.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



## 2.54.2 Feed

No feed support.

## 2.54.3 Link relations

Link relation	Description	Resource reference
self	This permissions resource	<a href="#">"Permissions" on page 503</a>

## 2.54.4 Operations

### 2.54.4.1 Supported HTTP methods

Method	Description
GET	GET a specified accessor's basic and extended permissions on a certain object

#### 2.54.4.2 Get the permission view instance

GET the computed effective basic and extended permissions on a specified SysObject for a specified accessor. Permission attributes are returned as embedded elements in the response message body.

##### 2.54.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

HTTP method	Query parameters	Description	Data type	Value range	Default value
	accessor	<p>A user name or group name to get permissions against for the SysObject.</p> <p>This parameter is not mandatory.</p> <p>The server returns the current login user's permissions by default.</p> <p>The server returns the system default permissions on this object when an unknown accessor (user) is provided by the client.</p>	string	N/A	The user that is currently logged in to the system

#### 2.54.4.2.2 Request media types

N/A

#### 2.54.4.2.3 Request headers

- Authorization
- Accept
- For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.54.4.2.4 Request body

N/A

#### 2.54.4.2.5 Response headers

- Content-Type
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.54.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.54.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.54.4.2.8 Response body

##### ➡ Example 2-203: XML request and response

```
GET http://localhost:8080/dctm-rest/repositories/  
REPO/objects/0900000580002507/permissions?accessor=dave  
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==  
Accept: application/vnd.emc.documentum+xml
```

Status Code 200 OK

Content-Type: application/vnd.emc.documentum+xml

```
<?xml version='1.0' encoding='UTF-8'?>  
<permission  
    xmlns="http://identifiers.emc.com/vocab/documentum"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" accessor="dave"  
    basic-permission="Relate" extend-  
    permissions="EXECUTE_PROC,CHANGE_LOCATION">  
    <links>  
        <link rel="self" href="http://localhost:8080/dctm-rest/repositories/  
REPO/  
        objects/0900000580002507/permissions?accessor=dave"/>  
    </links>  
</permission>
```



▶ **Example 2-204: JSON request and response**

```
GET http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580002507/
    permissions?accessor=dave
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
Accept: application/vnd.emc.documentum+json
```

Status Code 200 OK

Content-Type: application/vnd.emc.documentum+json

```
{
  "accessor": "dave",
  "basic-permission": "Relate",
  "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION",
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000580002507/permissions?accessor=dave"
    }
  ]
}
```



## 2.55 Permission Set

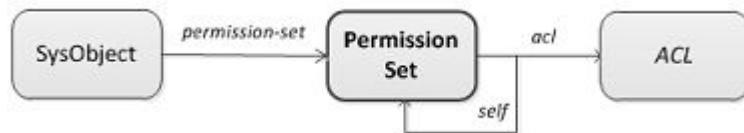
The Permission Set resource represents the permission set instance of a specified SysObject. Object level permission management, such as granting permissions or revoking permissions, is done using this resource's operations.

A permission set contains the following four sections that manage the access control of an object by using various aspects:

- *permitted*: This access level defines a set of users who are permitted to access the object.
- *restricted*: This access level defines a set of users who have restricted access to the object.
- *required group*: This access level requires that the user requesting access to the object must be a member of the group identified in the entry. If more than one group is defined in this section, the user requesting access to the object must be a member of all the groups defined.
- *required group set*: This access level requires that the user requesting access to the object be a member of at least one group in the set of groups defined.

### 2.55.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “[SysObject](#)” on page 692 and [About the Diagram](#) on page 13.

### 2.55.2 Feed

No feed support.

### 2.55.3 Link relations

The following table lists link relations for the Permission Set resource.

Link relation	Description	Resource reference
self	This Permission Set resource.	<a href="#">“Permission Set” on page 507</a>
edit	Edit this Permission Set resource.	<a href="#">“Permission Set” on page 507</a>
acl [1]	Retrieve the ACL object associated to this permission set object.	<a href="#">“ACL” on page 37</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <http://identifiers.emc.com/linkrel1/>.

### 2.55.4 Operations

#### 2.55.4.1 Supported HTTP methods

Method	Description
GET	GET a permission set
PUT	Update a permission set

#### 2.55.4.2 Get the Permission Set instance

GET the definition of the permission set of a specific object. The permission set returned may contain one or more sections of the complete permission set, which includes the *<permitted>*, *<restricted>*, *<required-group>*, and *<required-group-set>* sections. Sections that do not appear in the permission set returned are not set by that permission set.

##### 2.55.4.2.1 HTTP Request method and query parameters

GET

This method supports the following Request query parameter:

- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.55.4.2.2 Request media types

N/A

##### 2.55.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.55.4.2.4 Request body

N/A

##### 2.55.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.55.4.2.6 Supported Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.55.4.2.7 Response status

- 200 - Permission set retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.55.4.2.8 Response body

##### Example 2-205: XML representation

Status Code: 200 OK

Content-Type: application/vnd.emc.documentum+xml

```
<?xml version='1.0' encoding='UTF-8'?>
<permission-set
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <permitted>
        <permission accessor="dm_world" basic-permission="Read"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        <permission accessor="dm_owner" basic-permission="Delete"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        .....
    </permitted>
    <restricted>
        <permission accessor="docu" basic-permission="Version"/>
        .....
    </restricted>
    <required-group>
        <group>{group1}</group>
        <group>{group2}</group>
        .....
    </required-group>
    <required-group-set>
        <group>{group3}</group>
        <group>{group4}</group>
        .....
    </required-group-set>
    <links>
        <link rel="self" href="{permissionSetResourceUri}" />
        <link rel="edit" href="{permissionSetResourceUri}" />
        <link rel="http://identifiers.emc.com/linkrel/acl"
            href="{aclResourceUri}" />
    </links>
</permission-set>
```



▶ **Example 2-206: JSON representation**

Status Code: 200 OK

Content-Type: application/vnd.emc.documentum+json

```
{
  "permitted": [
    {
      "accessor": "dm_world",
      "basic-permission": "Read",
      "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION"
    },
    {
      "accessor": "dm_owner",
      "basic-permission": "Delete",
      "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION"
    },
    .....
  ],
  "restricted": [
    {
      "accessor": "docu",
      "basic-permission": "Version"
    }
    .....
  ],
  "required-group": [
    "{group1}",
    "{group2}"
    .....
  ],
  "required-group-set": [
    "{group3}",
    "{group4}"
    .....
  ],
  "links": [
    {
      "rel": "self",
      "href": "{permissionSetResourceUri}"
    },
    {
      "rel": "edit",
      "href": "{permissionSetResourceUri}"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/acl",
      "href": "{aclResourceUri}"
    }
  ]
}
```



#### 2.55.4.3 Update the Permission Set instance

The update operation on a permission set object enables you to do the following:

- grant or revoke a specific access level, or edit the existing access level for a user requesting access to the object (corresponding to add/delete/edit a user permission in the *<permitted>* attribute)
- add or remove a specific restricted level or edit the existing restricted level for a certain user that is requesting access to the object (corresponding to add/delete/edit a user permission in *restricted* attribute)
- add or remove a membership constraint to or from the user requesting access to the object (corresponding to add/delete a group in *required-group* or *required-group-set* attribute)

When the object permission set is updated, a newly created ACL is assigned to the specified object.

Anyone who has *<BROWSE>* basic permissions and *<CHANGE\_PERMIT>* extended permissions can update the basic permission. However, only the owner of the SysObject and the superuser can change basic and extended permissions. This means that because the PUT method must be used to update a permission set, only the owner of the SysObject and the superuser have sufficient permissions to update a permission set.

There are two built-in accessors: *dm\_world* and *dm\_owner*. Regardless of whether or not the REST client has specified any access level for *dm\_world* or *dm\_owner*, Documentum CM Server automatically grants users default permissions for the permission set.

The default basic and extended permission levels for these two users are defined in the *<default\_acl>* property of the server configuration object.

When updating a permission set, you must provide a complete permission set object in the Request body, rather than specifying only the part you want to update. When any property is null or empty, the property's value will be reset to null.

A permission set object may contain a list of user permissions. A user permission is made up of an accessor, a basic-permission, extend-permissions, and application-permission. The following table shows permission set and user permission object attributes:

Attribute	Description	Valid value	Default value	Mandatory?
accessor	A user or group name	Any valid user or group name		Yes  This is mandatory for a user permission

Attribute	Description	Valid value	Default value	Mandatory?
basic-permission	The basic permissions for an accessor	One of the following: <ul style="list-style-type: none"> <li>• None</li> <li>• Browse</li> <li>• Read</li> <li>• Relate</li> <li>• Version</li> <li>• Write</li> <li>• Delete</li> </ul>	None	No
extend-permissions	The extended permissions for an accessor	One or more of the following: <ul style="list-style-type: none"> <li>• EXECUTE_PROC</li> <li>• CHANGE_LOCATION</li> <li>• CHANGE_PERMISSION</li> <li>• CHANGE_STATE</li> <li>• DELETE_OBJECT</li> <li>• CHANGE_FOLDER_LINKS</li> </ul>	EXECUTE_PROC CHANGE_LOCATION	No
application-permission	Specifies a permission that is recognized by an application. An application permit is not recognized or enforced by Documentum CM Server. The application is responsible for enforcing the permission	string	NULL	No
permitted	A list of granted user permissions	User permissions	dm_world and dm_owner user permissions	No
restricted	A list of restricted user permissions	User permissions	NULL	No

Attribute	Description	Valid value	Default value	Mandatory?
required-group	A list of group names	Any valid group name	NULL	No
required-group-set	A list of group names	Any valid group name	NULL	No

#### 2.55.4.3.1 HTTP Request method and query parameters

PUT

#### 2.55.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.55.4.3.3 Request headers

- Accept
- Authorization
- Content-Type
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.55.4.3.4 Request body

##### Example 2-207: XML request

```
<permission-set
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <permitted>
        <permission accessor="dm_world" basic-permission="Read"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        <permission accessor="dm_owner" basic-permission="Write"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        .....
    </permitted>
    <restricted>
        ...
    </restricted>
    <required-group>
        ...
    </required-group>
    <required-group-set>
        ...
    </required-group-set>
</permission-set>
```



▶ **Example 2-208: JSON request**

```
{
    "permitted": [
        {
            "accessor": "dm_world",
            "basic-permission": "Read",
            "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION"
        },
        {
            "accessor": "dm_owner",
            "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION",
            "basic-permission": "Write"
        },
        .....
    ],
    "restricted": [
        .....
    ],
    "required-group": [
        .....
    ],
    "required-group-set": [
        .....
    ]
}
```



#### 2.55.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.55.4.3.6 Response media types

- application/vnd.emc.documentum+xml  
application/xml
- application/vnd.emc.documentum+json  
application/json

#### 2.55.4.3.7 Response status

- 200 - Permission set updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see Appendix C, REST Common Definition - HTTP Status Codes on page 1053.

#### 2.55.4.3.8 Response body

► **Example 2-209: XML response**

Status Code: 200 OK

Content-Type: application/vnd.emc.documentum+xml

```
<permission-set
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <permitted>
        <permission accessor="dm_world" basic-permission="Read"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        <permission accessor="dm_owner" basic-permission="Write"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        .....
    </permitted>
    <restricted>
        ...
    </restricted>
    <required-group>
        ...
    </required-group>
    <required-group-set>
        ...
    </required-group-set>
    <links>
        <link rel="self" href="{permissionSetResourceUri}" />
        <link rel="edit" href="{permissionSetResourceUri}" />
        <link rel="http://identifiers.emc.com/linkrel/acl"
            href="{aclResourceUri}" />
    </links>
</permission-set>
```



► **Example 2-210: JSON representation**

Status Code: 200 OK

Content-Type: application/vnd.emc.documentum+json

```
{
    "permitted": [
        {
            "accessor": "dm_world",
            "basic-permission": "Read",
            "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION"
        },
        {
            ...
        }
    ]
}
```

```

        "accessor": "dm_owner",
        "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION",
        "basic-permission": "Write"
    },
    .....
],
"restricted": [
    .....
],
"required-group": [
    .....
],
"required-group-set": [
    .....
],
"links": [
    {
        "rel": "self",
        "href": "{permissionSetResourceUri}"
    },
    {
        "rel": "edit",
        "href": "{permissionSetResourceUri}"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/acl",
        "href": "{aclResourceUri}"
    }
]
}

```

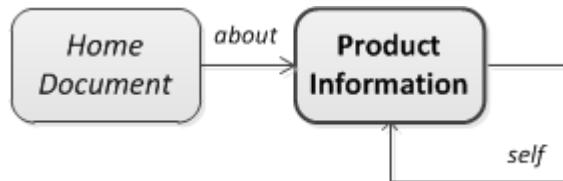


## 2.56 Product Information

The Product Information resource provides general information about the product, including the product name, major version, minor version, and so on.

### 2.56.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Home Document” on page 333 and About the Diagram on page 13.

## 2.56.2 Feed

No feed support.

## 2.56.3 Link relations

Link relation	Description	Resource reference
self	Production information	<a href="#">"Product Information" on page 517</a>

## 2.56.4 Operations

### 2.56.4.1 Supported HTTP methods

Method	Description
GET	Retrieves product information.

#### 2.56.4.2 Get Product Information

Retrieve product information about Foundation REST API, including product name, product version information. No authentication is required to perform this operation.

##### 2.56.4.2.1 HTTP Request method and query parameters

GET

##### 2.56.4.2.2 Request media types

N/A

##### 2.56.4.2.3 Request headers

- Accept



**Note:** The authorization header is not required when a client tries to get the Product Information resource.

#### 2.56.4.2.4 Request body

N/A

#### 2.56.4.2.5 Response headers

- Content-Type
- Content-Length

#### 2.56.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.56.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.56.4.2.8 Response body

XML or JSON representation of product information.

##### ➡ Example 2-211: XML response

```
<product-info name="documentum-rest-services-product-info">
  <properties>
    <product>Documentum REST Services</product>
    <product_version>7.3.xxx.yyy</product_version>
    <major>7.3</major>
    <minor>xxx</minor>
    <build_number>yyy</build_number>
    <revision_number>zzz</revision_number>
  </properties>
  <links>
    <link rel="self" href=
      "http://core-rs-demo.lss.emc.com:8080/dctm-rest/product-info" />
  </links>
</product-info>
```



##### ➡ Example 2-212: JSON response

```
{
  "name": "documentum-rest-services-product-info",
  "properties": {
    "product": "Documentum REST Services",
```

```
        "product_version": "7.3.xxx.yyy",
        "major": "7.3",
        "minor": "xxx",
        "build_number": "yyy",
        "revision_number": "zzz"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/product-
info.json"
        }
    ]
}
```



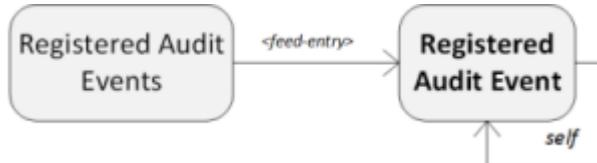
## 2.57 Registered Audit Event (s)

### 2.57.1 Registered Audit Event

The Registered Audit Event resource represents a single registered audit event for a specified object, folder, format, or repository. Logged in users must have CONFIG privileges added to their extended user permissions to get access to this resource.

#### 2.57.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Registered Audit Events](#)” on page 524 and [About the Diagram](#) on page 13.

#### 2.57.1.2 Feed

No feed support.

### 2.57.1.3 Link relations

Link relations	Description	Resource reference
self	This resource.	"Registered Audit Event" on page 520
delete	Used to unregister the specified event from the specified object.	"Registered Audit Event" on page 520

### 2.57.1.4 Operations

#### 2.57.1.4.1 Supported HTTP methods

HTTP method	Description
GET	GET a single registered audit event.
DELETE	Unregister a single audit event.

#### 2.57.1.4.2 Get a Registered Event

This operation retrieves a registered audit event from a specified object.

##### 2.57.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.57.1.4.2.2 Request media types

N/A

##### 2.57.1.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

**2.57.1.4.2.4 Request body**

N/A

**2.57.1.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.57.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.57.1.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.57.1.4.2.8 Response body** **Example 2-213: XML**

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-event xmlns="http://identifiers.emc.com/vocab/documentum"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <properties>
        <r_object_id>2600000580000505</r_object_id>
        <registered_id>0300000580000105</registered_id>
        <event>dm_save</event>
        <audit_subtypes>true</audit_subtypes>
        <controlling_app></controlling_app>
        <policy_id>0000000000000000</policy_id>
        <policy_state></policy_state>
        <a_authentication>0</a_authentication>
        <audit_attr_names xsi:nil="true"/>
        <sign_audit>false</sign_audit>
        <a_esignature_required>0</a_esignature_required>
        <user_name>dm_fulltext_index_user_01</user_name>
    </properties>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
                    registered-audit-events/2600000580000505" />
        <link rel="http://identifiers.emc.com/linkrel/delete"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
                    registered-audit-events/2600000580000505" />
    </links>
</audit-event>
```



#### 2.57.1.4.3 Delete a registered event

This operation unregisters a registered audit event.

##### 2.57.1.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.57.1.4.3.2 Request media types

N/A

##### 2.57.1.4.3.3 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.57.1.4.3.4 Request body

N/A

##### 2.57.1.4.3.5 Response headers

N/A

##### 2.57.1.4.3.6 Response media types

N/A

##### 2.57.1.4.3.7 Response status

- 204 - Unregistered successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.57.1.4.3.8 Response body

N/A

### 2.57.2 Registered Audit Events

The Registered Audit Events represents the registered audit events for a specified object, folder, format, or repository. Logged in users must have CONFIG privileges added to their extended user permissions to get access to this resource.

#### 2.57.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and [About the Diagram on page 13](#).

#### 2.57.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Audit Events
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: RegisteredAuditEvent
- Entry ID: URI of the Registered Audit Event
- Entry Title: The id of the specified registered event.
- Entry Updated: Current server time.

### 2.57.2.3 Link relations

Link relations	Description	Resource reference
self	This resource.	"Registered Audit Events" on page 524
unregister-all-audit-events [1]	Unregisters all registered audit events for the specified object.	"Registered Audit Events" on page 524
first, last, next and previous	Pagination links	"Registered Audit Events" on page 524
available-audit-events [1]	Retrieves the available audit event in a specified condition.	"Available Audit Events" on page 106
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

### 2.57.2.4 Operations

#### 2.57.2.4.1 Supported HTTP methods

HTTP methods	Descriptions
GET	Retrieves all of the registered events for the specified path requested.
POST	Registers an audit event for a specified object.
DELETE	Unregisters all audit events according to the query parameters set.

#### 2.57.2.4.2 Get All Registered Audit Events

##### 2.57.2.4.2.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• inline</li> <li>• page</li> <li>• items-per-page</li> <li>• include-total</li> <li>• filter</li> <li>• links</li> <li>• sort</li> <li>• view</li> </ul>	For more information, see Appendix B, REST Common Definition - URI Request Query Parameters on page 1047.			

HTTP methods	Query parameters	Description	Data type	Value range	Default value
	audit-scope	Defines the scope of impact for the audit events	String	One of the following values: <ul style="list-style-type: none"><li>• repository</li><li>• type</li><li>• object</li></ul>	null
	object-id	< <i>r_object_id</i> > of any object in Documentum CM Server. This parameter is meant to be used alone or as part of a statement.	String	Any valid < <i>r_object_id</i> > value.	null
	object-type	The name of the < <i>dm_type</i> > in Documentum CM Server. This parameter is meant to be used alone or as part of a statement.	String	Any valid < <i>dm_type</i> > name.	null

#### 2.57.2.4.2.2 Request media types

N/A

#### 2.57.2.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.57.2.4.2.4 Request body**

N/A

**2.57.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.57.2.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.57.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.57.2.4.2.8 Response body**

 **Example 2-214: XML**

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/registered-audit-
events</id>
  <title>Audit Events</title>
  <author>
    <name>EMC Documentum</name>
  </author>
  <updated>2017-05-24T03:05:05.381+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
  </dm:items-per-page>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
              registered-audit-events" />
  <link rel="http://identifiers.emc.com/linkrel/unregister-all-audit-events"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
              registered-audit-events" />
  <link rel="http://identifiers.emc.com/linkrel/available-audit-events"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
              available-audit-events" />
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/
```

```

        registered-audit-events/2600000580000905</id>
    <title>2600000580000905</title>
    <author>
        <name>auditadmin</name>
        <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
auditadmin</uri>
    </author>
    <summary>Detail of Registered Audit Event</summary>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/
registered-audit-events/2600000580000905"/>
    <content>
        <dm:audit-event xmlns:dm="http://identifiers.emc.com/vocab/
documentum"/>
    </content>
    </entry>
</feed>
```



### Example 2-215: JSON

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/registered-audit-
events",
    "title": "Audit Events",
    "author": [
        {
            "name": "EMC Documentum"
        }
    ],
    "updated": "2017-05-24T03:03:48.854+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
registered-audit-events"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/unregister-all-audit-events",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
registered-audit-events"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/available-audit-events",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
available-audit-events"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/
registered-audit-events/2600000580000905",
            "title": "2600000580000905",
            "author": [
                {
                    "name": "auditadmin",
                    "uri": "http://localhost:8080/dctm-rest/repositories/REPO/users/
auditadmin"
                }
            ]
        }
    ]
}
```

```

        }
    ],
    "summary": "Detail of Registered Audit Event",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                registered-audit-events/2600000580000905"
        }
    ],
    "content": {
    }
}
]
}

```



#### 2.57.2.4.3 Register an audit event

##### 2.57.2.4.3.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
POST	audit-scope <i>Mandatory query parameter</i>	Defines the scope of impact for the audit events	String	One of the following values: <ul style="list-style-type: none"><li>• repository</li><li>• type</li><li>• object</li></ul>	null
	object-id	< <i>r_object_id</i> > of any object in Documentum CM Server. This parameter is meant to be used as part of a statement.	String	Any valid < <i>r_object_id</i> > value.	null

HTTP methods	Query parameters	Description	Data type	Value range	Default value
	object-type	The name of the <i>&lt;dm_type&gt;</i> in Documentum CM Server. This parameter is meant to be used as part of a statement.	String	Any valid <i>&lt;dm_type&gt;</i> name.	null



### Caution

The object-id query parameter is mandatory when used as part of this statement.*<audit-scope=object>*

This object-type query parameter is mandatory when used as part of this statement.*<audit-scope=type>*

#### 2.57.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.57.2.4.3.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.57.2.4.3.4 Request body



##### Example 2-216: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<register-audit-event xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_register-audit-event">
    <properties>
        <event>dm_getlogin</event>
    </properties>
</register-audit-event>
```



 **Example 2-217: JSON**

```
{
  "properties": {
    "event": "dm_getlogin",
  }
}
```



#### 2.57.2.4.3.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.57.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.57.2.4.3.7 Response status

- 201 - Registered successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.57.2.4.3.8 Response body

 **Example 2-218: XML**

```
<?xml version='1.0' encoding='UTF-8'?>
<audit-event xmlns="http://identifiers.emc.com/vocab/documentum"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:type="dmi_registry">
  <properties>
    <user_name>auditadmin</user_name>
    <registered_id>0000000000000000</registered_id>
    <event>dm_save</event>
    <oneshot>false</oneshot>
    <sendmail>false</sendmail>
    <priority>0</priority>
    <message>Save Object</message>
    <is_audittrail>true</is_audittrail>
    <policy_id>0000000000000000</policy_id>
    <audit_subtypes>false</audit_subtypes>
    <a_authentication>0</a_authentication>
```

```

<sign_audit>false</sign_audit>
<a_esignature_required>0</a_esignature_required>
<i_is_replica>false</i_is_replica>
<i_vstamp>0</i_vstamp>
<r_object_id>2600000580000906</r_object_id>
</properties>
<links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
            registered-audit-events?audit-scope=repository"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
            registered-audit-events/2600000580000906"/>
</links>
</audit-event>

```



#### ➡ Example 2-219: JSON

```
{
  "json-root": "audit-event",
  "type": "dmi_registry",
  "properties": {
    "user_name": "auditadmin",
    "registered_id": "0000000000000000",
    "event": "dm_save",
    "oneshot": false,
    "sendmail": false,
    "priority": 0,
    "message": "Save Object",
    "is_audittrail": true,
    "policy_id": "0000000000000000",
    "audit_subtypes": false,
    "a_authentication": 0,
    "sign_audit": false,
    "a_esignature_required": 0,
    "i_is_replica": false,
    "i_vstamp": 0,
    "r_object_id": "2600000580000906"
  },
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
          registered-audit-events?audit-scope=repository"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/delete",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
          registered-audit-events/2600000580000906"
    }
  ]
}
```



 **Notes**

- The value of `<controlling_app>` is stored in Documentum CM Server in lowercase.
- The `<policy_id>`, `<policy_state>`, and `<audit_subtypes>` parameters are not permitted when performing an audit on an object instance.

#### 2.57.2.4.4 Unregister all registered audit events

##### 2.57.2.4.4.1 HTTP Request method and query parameters

HTTP methods	Query parameters	Description	Data type	Value range	Default value
DELETE	audit-scope <i>Mandatory query parameter</i>	Defines the scope of impact for the audit events	String	One of the following values: <ul style="list-style-type: none"><li>• repository</li><li>• type</li><li>• object</li></ul>	null
	object-id	<code>&lt;r_object_id&gt;</code> of any object in Documentum CM Server. This parameter is meant to be used as part of a statement.	String	Any valid <code>&lt;r_object_id&gt;</code> value.	null
	object-type	The name of the <code>&lt;dm_type&gt;</code> in Documentum CM Server. This parameter is meant to be used as part of a statement.	String	Any valid <code>&lt;dm_type&gt;</code> name.	null


**Caution**

The object-id query parameter is mandatory when used as part of this statement. `<audit-scope=object>`

The object-type query parameter is mandatory when used as part of this statement. `<audit-scope=type>`

#### **2.57.2.4.4.2 Request media types**

N/A

#### **2.57.2.4.4.3 Request headers**

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.57.2.4.4.4 Request body**

N/A

#### **2.57.2.4.4.5 Response headers**

N/A

#### **2.57.2.4.4.6 Response media types**

N/A

#### **2.57.2.4.4.7 Response status**

- 204 - Unregister procedure succeeded
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.57.2.4.4.8 Response body**

N/A

#### **2.57.2.4.4.9 Additional information on events**

Documentum CM Server audits some events by default. For more information, see [Appendix F, Events Audited by Default on page 1077](#).

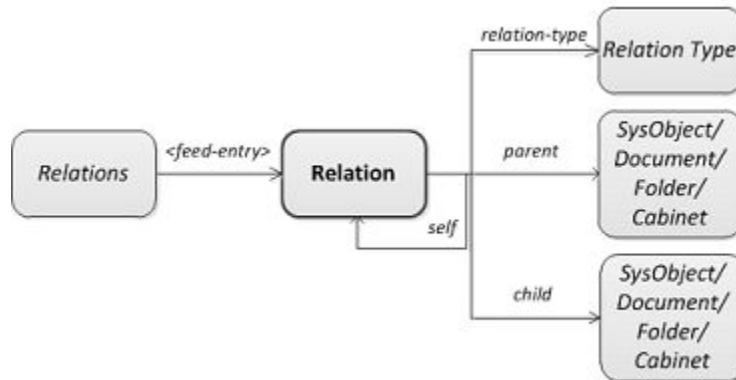
## **2.58 Relation (s)**

## 2.58.1 Relation

The Relation resource represents a single relation instance in a repository.

### 2.58.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Relations” on page 538 and About the Diagram on page 13.

### 2.58.1.2 Feed

No feed support.

### 2.58.1.3 Link relations

Link relation	Description	Resource reference
self	This Relation resource	“Relation” on page 535
relation-type [1]	Relation type definition	“Relation Type” on page 556
parent	Reference to the parent object of this instance	“SysObject” on page 692 “Document” on page 214 “Folder” on page 233 “Cabinet” on page 132
child	Reference to the child object of this instance	“SysObject” on page 692 “Document” on page 214 “Folder” on page 233 “Cabinet” on page 132

Link relation	Description	Resource reference
	[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .	

## 2.58.1.4 Operations

### 2.58.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a specified relation.
DELETE	Deletes a specified relation.

### 2.58.1.4.2 Get a Relation

Retrieve a specified relation instance.

#### 2.58.1.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameter:

HTTP method	Query parameter	Description	Data type	Value range	Default value
GET	view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.58.1.4.2.2 Request media types

N/A

#### 2.58.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.58.1.4.2.4 Request body**

N/A

**2.58.1.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.58.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

**2.58.1.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.58.1.4.2.8 Response body**

An XML or JSON representation of the relation.

**2.58.1.4.3 Delete a Relation**

Delete a specified relation in the repository.

**2.58.1.4.3.1 HTTP Request method and query parameters**

DELETE

**2.58.1.4.3.2 Request media types**

N/A

#### 2.58.1.4.3.3 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.58.1.4.3.4 Request body

N/A

#### 2.58.1.4.3.5 Response headers

N/A

#### 2.58.1.4.3.6 Response media types

N/A

#### 2.58.1.4.3.7 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.58.1.4.3.8 Response body

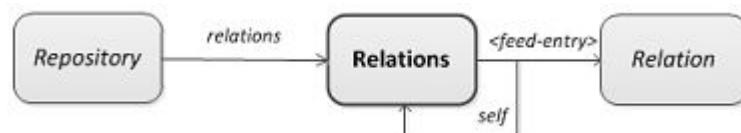
HTTP 204 No Content status upon a successful delete operation. The response body contains no content.

## 2.58.2 Relations

The Relations resource represents a collection of relations in a repository.

### 2.58.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram on page 13](#).

### 2.58.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of relations
- Supports POST: Yes
- Entry: “[Relation](#)” on page 535
- Entry ID: URI of the relation
- Entry Title: Relation name
- Entry Summary: Relation description
- Entry Updated: `r_modify_date` of the relation

### 2.58.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of relations.	“ <a href="#">Relations</a> ” on page 538
first, last, next, previous	Pagination links.	“ <a href="#">Relations</a> ” on page 538

### 2.58.2.4 Operations

#### 2.58.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the information about a collection of relations.
POST	Create a new relation instance between two objects.

#### 2.58.2.4.2 Get Relations

Retrieve the information about a collection of relations.

##### 2.58.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	relation-name	Only relations whose name equals to the specified string are returned in the result.  Relation names are case sensitive.	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	related-object-id	Only relations that contain an object whose object ID equals to the specified string are returned in the result.	string	N/A	null
	related-object-role	Indicates the role of the object specified by the object ID. (Only works when the related-object-id parameter is specified.)  Valid values are: <ul style="list-style-type: none"><li>• child</li><li>• parent</li><li>• any</li></ul>	string	N/A	<any>

This method supports the following common query parameters:

- inline
- view
- items-per-page
- page
- include-total
- sort
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.58.2.4.2.2 Request media types

N/A

#### 2.58.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.58.2.4.2.4 Request body

N/A

#### 2.58.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.58.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.58.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.58.2.4.2.8 Response body

XML or JSON representation of the collection of relations.

- The body contains a list of the `dm_relation` instances (or subtypes of `dm_relation`).
- Each object may contain all or a set of properties of the relation, depending on the setting of the query parameter `view`.
- Pagination is supported.
- By default, the results are listed in alphabetical order by relation name.

## 2.59 Related Documents

The *Related Documents* resource forms part of the related document service of the Documentum Contextual Content Engine. This resource can find related or similar documents using common values among documents, such as keywords.

### 2.59.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Document](#)” on page 214 and [About the Diagram](#) on page 13.

### 2.59.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
- Feed Title: Related Documents of <document name> from CCE
- Supports POST: No
- Feed Updated: Current server time
- Entry: “[Document](#)” on page 214
- Entry ID: URI of the Document resource
- Entry Title: The name of the document
- Entry Updated: Current server time

### 2.59.3 Link relations

Link relation	Description	Resource reference
self	This collection of related documents.	“ <a href="#">Related Documents</a> ” on page 542

## 2.59.4 Operations

### 2.59.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the related documents of the requested document.

### 2.59.4.2 Get Related Documents

Retrieves the related documents of the requested document.

#### 2.59.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.59.4.2.2 Request media types

N/A

#### 2.59.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.59.4.2.4 Request body

N/A

#### 2.59.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.59.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json
- application/hal+json

#### 2.59.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.59.4.2.8 Response body

##### ➡ Example 2-220: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
      090000018000115c/related-docs</id>
  <title>Related Documents of document 090000018000115c from CCE</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2018-08-20T02:34:35.756+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    1
  </dm:page>
  <dm:items-per-page
      xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
  </dm:items-per-page>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/documents/
            090000018000115c/related-docs"></link>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
        090000018000115a</id>
    <title>Laboratory-Manager</title>
    <author>
      <name>Administrator</name>
      <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
          Administrator</uri>
```

```

        </author>
        <summary>dm_document 090000018000115a</summary>
        <updated>2018-08-01T02:48:54.000+00:00</updated>
        <published>2018-07-31T09:16:35.000+00:00</published>
        <link rel="edit"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
              documents/090000018000115a"></link>
        <content type="application/xml"
                  src="http://localhost:8080/dctm-rest/repositories/REPO/
                  documents/090000018000115a"></content>
    </entry>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
          090000018000115d</id>
        <title>Project-Manager</title>
        <author>
            <name>Administrator</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
              Administrator</uri>
        </author>
        <author>
            <name>mary</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
              mary</uri>
        </author>
        <author>
            <name>dave</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
              dave</uri>
        </author>
        <summary>dm_document 090000018000115d</summary>
        <updated>2018-08-01T02:59:30.000+00:00</updated>
        <published>2018-07-31T09:19:08.000+00:00</published>
        <link rel="edit"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
              documents/090000018000115d"></link>
        <content type="application/xml"
                  src="http://localhost:8080/dctm-rest/repositories/REPO/
                  documents/090000018000115d"></content>
    </entry>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
          090000018000115b</id>
        <title>Product-Manager</title>
        <author>
            <name>mary</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
              mary</uri>
        </author>
        <author>
            <name>dave</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
              dave</uri>
        </author>
        <author>
            <name>Administrator</name>
            <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/
              Administrator</uri>
        </author>
        <summary>dm_document 090000018000115b</summary>
        <updated>2018-08-01T02:53:37.000+00:00</updated>
        <published>2018-07-31T09:17:01.000+00:00</published>
    </entry>

```

```

<link rel="edit"
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            documents/090000018000115b"></link>
<content type="application/xml"
          src="http://localhost:8080/dctm-rest/repositories/REPO/
                documents/090000018000115b"></content>
</entry>
</feed>
```



### Example 2-221: JSON Response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
        090000018000115c/related-docs",
  "title": "Related Documents of document 090000018000115c from CCE",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2018-08-20T02:30:30.334+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                documents/090000018000115c/related-docs"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/
            documents/090000018000115a",
      "title": "Laboratory-Manager",
      "author": [
        {
          "name": "Administrator",
          "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/Administrator"
        }
      ],
      "summary": "dm_document 090000018000115a",
      "updated": "2018-08-01T02:48:54.000+00:00",
      "published": "2018-07-31T09:16:35.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/
                    REPO/documents/090000018000115a"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                  documents/090000018000115a"
      }
    },
    {

```

```

        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
              documents/090000018000115d",
        "title": "Project-Manager",
        "author": [
            {
                "name": "Administrator",
                "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                      users/Administrator"
            },
            {
                "name": "mary",
                "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                      users/mary"
            },
            {
                "name": "dave",
                "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                      users/dave"
            }
        ],
        "summary": "dm_document 090000018000115d",
        "updated": "2018-08-01T02:59:30.000+00:00",
        "published": "2018-07-31T09:19:08.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/repositories/
                      REPO/documents/090000018000115d"
            }
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                  documents/090000018000115d"
        }
    },
    {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
              documents/090000018000115b",
        "title": "Product-Manager",
        "author": [
            {
                "name": "mary",
                "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                      users/mary"
            },
            {
                "name": "dave",
                "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                      users/dave"
            },
            {
                "name": "Administrator",
                "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
                      users/Administrator"
            }
        ],
        "summary": "dm_document 090000018000115b",
        "updated": "2018-08-01T02:53:37.000+00:00",
        "published": "2018-07-31T09:17:01.000+00:00",
        "links": [
            {

```

```
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/documents/090000018000115b"
    }
],
"content": {
    "type": "application/vnd.emc.documentum+json",
    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115b"
}
}
]
```



#### ➡ Example 2-222: HAL: JSON Response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
090000018000115c/related-docs",
    "title": "Related Documents of document 090000018000115c from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-08-20T02:30:30.334+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115c/related-docs"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115a",
            "title": "Laboratory-Manager",
            "author": [
                {
                    "name": "Administrator",
                    "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
users/Administrator"
                }
            ],
            "summary": "dm_document 090000018000115a",
            "updated": "2018-08-01T02:48:54.000+00:00",
            "published": "2018-07-31T09:16:35.000+00:00",
            "links": [
                {
                    "rel": "edit",
                    "href": "http://localhost:8080/dctm-rest/repositories/
REPO/documents/090000018000115a"
                }
            ],
            "content": {
                "type": "application/vnd.emc.documentum+json",

```

```

        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115a"
    }
},
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115d",
    "title": "Project-Manager",
    "author": [
        {
            "name": "Administrator",
            "uri": "http://localhost:8080/dctm-rest/repositories/
REPO/users/Administrator"
        },
        {
            "name": "mary",
            "uri": "http://localhost:8080/dctm-rest/repositories/
REPO/users/mary"
        },
        {
            "name": "dave",
            "uri": "http://localhost:8080/dctm-rest/repositories/
REPO/users/dave"
        }
    ],
    "summary": "dm_document 090000018000115d",
    "updated": "2018-08-01T02:59:30.000+00:00",
    "published": "2018-07-31T09:19:08.000+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/
REPO/documents/090000018000115d"
        }
    ],
    "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115d"
    }
},
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115b",
    "title": "Product-Manager",
    "author": [
        {
            "name": "mary",
            "uri": "http://localhost:8080/dctm-rest/repositories/
REPO/users/mary"
        },
        {
            "name": "dave",
            "uri": "http://localhost:8080/dctm-rest/repositories/
REPO/users/dave"
        },
        {
            "name": "Administrator",
            "uri": "http://localhost:8080/dctm-rest/repositories/
REPO/users/Administrator"
        }
    ],
}
]

```

```
"summary": "dm_document 090000018000115b",
"updated": "2018-08-01T02:53:37.000+00:00",
"published": "2018-07-31T09:17:01.000+00:00",
"links": [
    {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/documents/090000018000115b"
    }
],
"content": {
    "type": "application/vnd.emc.documentum+json",
    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115b"
```



## 2.60 Related SME

The *Related Subject Matter Experts (SME)* resource forms part of the services of the Documentum Contextual Content Engine. This resource can find related SME (Authors) who have done work on a given document.

## 2.60.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Document” on page 214 and About the Diagram on page 13.

## 2.60.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
  - Feed Title: Related SME (authors) of <document name> from CCE
  - Supports POST: No
  - Feed Updated: Current server time
  - Entry: “User” on page 766
  - Entry ID: The User resource URI
  - Entry Title: The name of the user

- Entry Updated: Current server time

### 2.60.3 Link relations

Link relation	Description	Resource reference
self	This collection of related SMEs.	"Related SME" on page 550

### 2.60.4 Operations

#### 2.60.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the related SME of the requested document.

#### 2.60.4.2 Get Related SMEs

Retrieves the related SME of the requested document.

##### 2.60.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, links	For more information, see Appendix B, REST Common Definition - URI Request Query Parameters on page 1047.			

For more information, see Appendix B, REST Common Definition - URI Request Query Parameters on page 1047.

##### 2.60.4.2.2 Request media types

N/A

##### 2.60.4.2.3 Request headers

- Accept
- Authorization

For more information, see Appendix D, REST Common Definition - HTTP Headers on page 1055.

**2.60.4.2.4 Request body**

N/A

**2.60.4.2.5 Response headers**

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.60.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json
- application/hal+json

**2.60.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.60.4.2.8 Response body** **Example 2-223: XML Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/090000018000115c/related-sme</id>
    <title>Related SMEs(authors) of document 090000018000115c from CCE</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2018-08-20T02:52:47.378+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">100
    </dm:items-per-page>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/documents/090000018000115c/related-sme"></link>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/users/dave</id>
        <title>dave</title>
        <summary></summary>
```

```

<updated>2018-08-01T02:28:30.000+00:00</updated>
<published>2018-08-01T02:28:30.000+00:00</published>
<link rel="edit"
      href="http://localhost:8080/dctm-rest/repositories/REPO/users/
            dave"></link>
<content type="application/xml"
          src="http://localhost:8080/dctm-rest/repositories/REPO/users/
            dave"></content>
</entry>
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/users/
        Administrator</id>
    <title>Administrator</title>
    <summary></summary>
    <updated>2018-07-31T03:56:09.000+00:00</updated>
    <published>2018-07-31T03:56:09.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/users/
            Administrator"></link>
    <content type="application/xml"
              src="http://localhost:8080/dctm-rest/repositories/REPO/users/
            Administrator"></content>
</entry>
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/users/mary</id>
    <title>mary</title>
    <summary></summary>
    <updated>2018-08-01T02:30:03.000+00:00</updated>
    <published>2018-08-01T02:30:03.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/users/
            mary"></link>
    <content type="application/xml"
              src="http://localhost:8080/dctm-rest/repositories/REPO/users/
            mary"></content>
</entry>
</feed>
```



#### Example 2-224: JSON Response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
          090000018000115c/related-sme",
    "title": "Related SMEs(authors) of document 090000018000115c from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-08-20T02:47:02.551+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                  documents/090000018000115c/related-sme"
        }
    ],
}
```

```
"entries": [
    {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/users/
              dave",
        "title": "dave",
        "summary": "",
        "updated": "2018-08-01T02:28:30.000+00:00",
        "published": "2018-08-01T02:28:30.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/repositories/
                      REPO/users/dave"
            }
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/dave"
        }
    },
    {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/users/
              Administrator",
        "title": "Administrator",
        "summary": "",
        "updated": "2018-07-31T03:56:09.000+00:00",
        "published": "2018-07-31T03:56:09.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/repositories/
                      REPO/users/Administrator"
            }
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/Administrator"
        }
    },
    {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/users/
              mary",
        "title": "mary",
        "summary": "",
        "updated": "2018-08-01T02:30:03.000+00:00",
        "published": "2018-08-01T02:30:03.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/repositories/
                      REPO/users/mary"
            }
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/mary"
        }
    }
]
```

```

    ]
}
```



### Example 2-225: HAL: JSON Response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
         090000018000115c/related-sme",
  "title": "Related SMEs(authors) of document 090000018000115c from CCE",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2018-09-14T02:07:01.300+00:00",
  "page": 1,
  "items-per-page": 100,
  "_embedded": {
    "collection": [
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/dave",
        "title": "dave",
        "updated": "2018-08-01T02:28:30.000+00:00",
        "published": "2018-08-01T02:28:30.000+00:00",
        "_links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
                     REPO/users/dave"
          }
        }
      },
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/Administrator",
        "title": "Administrator",
        "updated": "2018-07-31T03:56:09.000+00:00",
        "published": "2018-07-31T03:56:09.000+00:00",
        "_links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
                     REPO/users/Administrator"
          }
        }
      },
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                  users/mary",
        "title": "mary",
        "updated": "2018-08-01T02:30:03.000+00:00",
        "published": "2018-08-01T02:30:03.000+00:00",
        "_links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
                     REPO/users/mary"
          }
        }
      }
    ]
}
```

```

},
"_links": {
  "self": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/documents/090000018000115c/related-sme"
  }
}
}

```



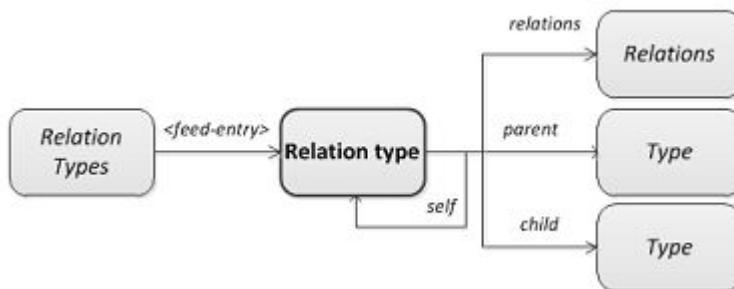
## 2.61 Relation Type (s)

### 2.61.1 Relation Type

The Relation Type resource represents a single instance of a relation type (`dm_relation_type` and its subtypes) in a repository.

#### 2.61.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Relation Types” on page 558 and About the Diagram on page 13.

#### 2.61.1.2 Feed

No feed support.

#### 2.61.1.3 Link relations

Link relation	Description	Resource reference
self	This Relation resource.	<a href="#">“Relation Type” on page 556</a>
relations [1]	Relations that use this relation type	<a href="#">“Relations” on page 538</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.61.1.4 Operations

### 2.61.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a specified Relation Type resource.

### 2.61.1.4.2 Get a Relation Type

Retrieve a specified Relation Type resource.

#### 2.61.1.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameter:

- view

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.61.1.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.61.1.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.61.1.4.2.4 Request body

N/A

#### 2.61.1.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.61.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.61.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.61.1.4.2.8 Response body

XML or JSON representation of the relation type.



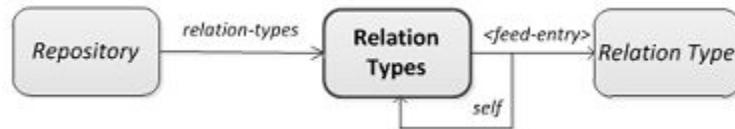
**Note:** The relation type name in the URL is encoded.

## 2.61.2 Relation Types

The Relation Types resource represents a collection of relation type (`dm_relation_type` and its subtypes) instances in a repository.

#### 2.61.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram on page 13](#).

### 2.61.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the relation types
- Feed Title: Relation Types
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: “[Relation Type](#)” on page 556
- Entry ID: URI of the relation type
- Entry Title: Relation type name
- Entry Summary: Relation type description
- Entry Updated: `r_modify_date` of the relation type

### 2.61.2.3 Link relations

Link relation	Description	Resource reference
self	This Relation Types resource.	“ <a href="#">Relation Types</a> ” on page 558
first, last, next, previous	Pagination links	“ <a href="#">Relation Types</a> ” on page 558

### 2.61.2.4 Operations

#### 2.61.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of Relation Type instances.

#### 2.61.2.4.2 Get Relation Types

Retrieve a collection of Relation Type instances.

##### 2.61.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	view	Defines the properties to return.	string	<p>The pattern is: &lt;? view=(:view-name)?(,column)* - (:view-name) and (,columns)*&gt; and must be mutually exclusive names of predefined views start with ":".</p> <p>The following view-names are defined:&lt;- :all&gt;&lt;- :default&gt;- if no view-name is specified, then a comma-separated list of the names of properties or predefined views are returned.</p>	<:default>
	links	Defines whether link relations to be returned for this object representation	string	<p>true or <i>false</i></p> <ul style="list-style-type: none"> <li>• &lt;links=true&gt; is default</li> <li>• &lt;links=false&gt; means "Do not send me any links that are too expensive to calculate. Only send me links that are cheap to calculate".</li> </ul>	<true>
	inline	Ensures whether to show content (the object instance) in the atom entry for a collection;	string	<ul style="list-style-type: none"> <li>• &lt;true&gt; - Returns the object instance and embeds an object instance into the entry's content element</li> <li>• &lt;false&gt; - Does not return an object instance.</li> </ul>	<false>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	sort	Defines sorting for entries in the collection result	string	<p>&lt;sort&gt; is an optional parameter that is created using a comma separated list of sort specifications that, when combined, create a sort order. Each sort specification consists of an attribute to be sorted and its sort order, separated by a space character.</p> <p>The sort order can be either “DESC” or “ASC”, and the sort can also be case insensitive.</p> <p>Optionally, sort order can be specified with non-repeating attributes;</p> <p>For example: sort=r_modify_date desc, object_id asc,title</p> <p>When an attribute with invalid name is specified, an error is thrown.</p>	The default value of this parameter may differ by resource
	filter	A filter expression in a subset of XPath.	filter expression	N/A	No filter is applied
	page	Defines the number of the page to be served.  When the result feed is paged, an <dm:page> xxx</dm:page> element is displayed under the feed root.	integer	= 1	<1>  For example: items_per_page=200&page=2 returns items 201 to 400

HTTP method	Query parameters	Description	Data type	Value range	Default value
	items-per-page	number of entries per page; when the result feed is paged, an element <dm:items-per-page>xxx</dm:items-per-page> will be displayed under the feed root	integer	>=1	<100>
	include-total	indicate to calculate the total count of the feed items as though not returning them all in one page; when it is <i>true</i> , an element <dm:total>xxx</dm:total> will be displayed under the feed root	string	<ul style="list-style-type: none"> <li>&lt;<i>true</i>&gt; - The server calculates the total count of feed items and returns that value.</li> <li>&lt;<i>false</i>&gt; - Does not calculate and return a total count.</li> </ul>	< <i>false</i> >

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

**2.61.2.4.2.2 Request media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

**2.61.2.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.61.2.4.2.4 Request body**

N/A

**2.61.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.61.2.4.2.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

**2.61.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.61.2.4.2.8 Response body

XML or JSON representation of the relation types.

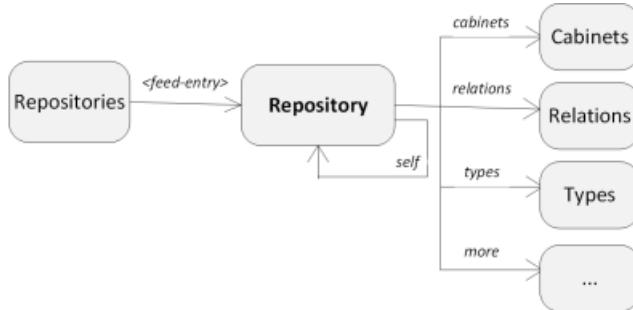
 **Note:** The type name in the URL is encoded.

## 2.62 Repository

The Repository resource provides information about a single repository. This resource requires authentication.

### 2.62.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repositories” on page 573 and [About the Diagram on page 13](#).

### 2.62.2 Feed

No feed support.

### 2.62.3 Link relations

Link relation	Description	Resource reference
acls [1]	Link to ACLs in this repository.	<a href="#">“ACLs Collection” on page 50</a>
archived-contents [1]	Link to archived contents resource	<a href="#">“Archived Contents” on page 65</a>
aspect-types [1]	Link to aspect types resource.	<a href="#">“Aspect Types” on page 78</a>
batches [1]	Link to batches resource.	<a href="#">“Batches” on page 110</a>
batch-capabilities [1]	Link to batch capabilities resource.	<a href="#">“Batch Capabilities” on page 128</a>
cabinets [1]	Link to the cabinets collection.	<a href="#">“Cabinets” on page 138</a>
checked-out-objects [1]	Link to the Checked Out Objects resource.	<a href="#">“Checked Out Objects” on page 143</a>

<b>Link relation</b>	<b>Description</b>	<b>Resource reference</b>
current-user [1]	Link to the current login user of the repository.	<a href="#">“Current User” on page 188</a>
current-user-preferences [1]	Link to the Current User Preferences resource.	<a href="#">“Current User Preferences” on page 198</a>
dql [1]	Link to the dql resource in this repository.	<a href="#">“DQL” on page 224</a>
formats [1]	Link to the formats in this repository.	<a href="#">“Formats” on page 305</a>
groups [1]	Link to the groups in this repository.	<a href="#">“Groups” on page 317</a>
lifecycles [1]	Link to the Lifecycles resource.	<a href="#">“Lifecycles” on page 385</a>
network-locations [1]	Link to the network locations of this repository.	<a href="#">“Network Locations” on page 449</a>
relations [1]	Link to the Relations resource.	<a href="#">“Relations” on page 538</a>
relation-types [1]	Link to the relation types in this repository.	<a href="#">“Relation Types” on page 558</a>
saved-searches [1]	Link to the saved-searches resource.	<a href="#">“Saved Searches” on page 586</a>
search [1]	Link to the search resource.	<a href="#">“Search” on page 645</a>
search-templates [1]	Link to search templates resource.	<a href="#">“Saved Search Templates” on page 621</a>
self	Link to this repository.	<a href="#">“Repository” on page 564</a>
snapshots	Link to the Virtual Document Snapshots resource.	<a href="#">“Virtual Document Snapshots” on page 839</a>
repository	Link to the repository instance itself.	<a href="#">“Repositories” on page 573</a>
types [1]	Link to the Types resource.	<a href="#">“Types” on page 755</a>
users [1]	Link to the users in this repository.	<a href="#">“Users” on page 775</a>
subscriptions [1]	You, as the current user, can view all subscriptions in the repository.	<a href="#">“User Subscriptions” on page 798</a>
word-suggestions [1]	Allows you to use word suggestions while searching for content with Documentum Contextual Content Engine.	<a href="#">“Word Suggestions” on page 863</a>
synonyms [1]	Allows you to use synonyms while searching for content with Documentum Contextual Content Engine.	<a href="#">“Synonyms” on page 682</a>

Link relation	Description	Resource reference
synonym-categories [1]	Allows you to use synonym-categories while searching for content with Documentum Contextual Content Engine.	" <a href="#">Synonym Categories</a> " on page 686
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.62.4 Operations

### 2.62.4.1 Supported HTTP methods

Method	Description
GET	Retrieves properties of the Repository resource.

### 2.62.4.2 Get a repository

Retrieves information about a single repository. Non super users cannot see name, version, and connection broker details.

The following restrictions are applicable in the /repositories/repository API response:

- Server section is omitted for users who are not administrators or super users.
- Server section is retained for administrators or super users, but only the host information in the server section is omitted.

#### 2.62.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			
	include-domains	Specifies whether to return domain names. When this parameter is set to <i>true</i> , the API response includes domain names.	string	<i>true, false</i>	<i>false</i>

#### 2.62.4.2.2 Request media types

N/A

#### 2.62.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.62.4.2.4 Request body

N/A

#### 2.62.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.62.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.62.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.62.4.2.8 Response body

XML or JSON representation of the repository. Foundation REST API version 16.4 and later, provide the following two attributes:

- auth-protocol: When this attribute is set to domain-required on Windows platforms, the repository runs in domain-required mode.

When this attribute is not set to domain-required, the repository does not run in domain-required mode and this property is blank.

When you are authenticating users on the UNIX platform against a Windows domain, you must set this attribute to unix\_domain\_used. Otherwise, it remains blank for repositories that run on the UNIX platform.

- domains: Retrieves a list of domain names that are in the repository. The domains attribute does not appear in the response when no domain has been configured in the repository.

### ➤ Example 2-226: XML response

```
<?xml version='1.0' encoding='utf-8'?>
<repository xmlns="http://identifiers.emc.com/vocab/documentum"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>12</id>
    <name>REPO1</name>
    <description></description>
    <auth-protocol></auth-protocol>
    <server>
        <name>REPO1</name>
        <host>CS73</host>
        <version xml:space="preserve">
            7.3.0000.0214 Win64.SQLServer
        </version>
        <docbroker>CS73</docbroker>
    </server>
    <domains>
        <domain>joli-146</domain>
        <domain>Copy of joli-146</domain>
    </domains>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/REPO1?
                  include-domains=true" />
        <link rel="http://identifiers.emc.com/linkrel/cabinets"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/cabinets" />
        <link rel="http://identifiers.emc.com/linkrel/checked-out-objects"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/checked-out-
              objects" />
        <link rel="http://identifiers.emc.com/linkrel/current-user"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/
              currentuser" />
        <link rel="http://identifiers.emc.com/linkrel/current-user-preferences"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/
              currentuser-preferences" />
        <link rel="http://identifiers.emc.com/linkrel/users"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/users" />
        <link rel="http://identifiers.emc.com/linkrel/groups"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/groups" />
        <link rel="http://identifiers.emc.com/linkrel/formats"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/format" />
        <link rel="http://identifiers.emc.com/linkrel/network-locations"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/network-
              locations" />
        <link rel="http://identifiers.emc.com/linkrel/relations"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/relations" />
        <link rel="http://identifiers.emc.com/linkrel/relation-types"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/relation-
              types" />
        <link rel="http://identifiers.emc.com/linkrel/types"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/types" />
        <link rel="http://identifiers.emc.com/linkrel/aspect-types"
              href="http://localhost:8080/dctm-rest/repositories/REPO1/aspect-
              types" />
        <link rel="http://identifiers.emc.com/linkrel/dql"
              hreftemplate="http://localhost:8080/dctm-rest/repositories/REPO1{?" />
```

```

dql}" />
    <link rel="http://identifiers.emc.com/linkrel/search"
      hreftemplate="http://localhost:8080/dctm-rest/repositories/REPO1/
      search{?
          collections,facet,include-total,inline,items-per-
          page,locations,
          object-type,page,q,sort,timezone,view}" />
    <link rel="http://identifiers.emc.com/linkrel/saved-searches"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/saved-
      searches" />
    <link rel="http://identifiers.emc.com/linkrel/search-templates"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/search-
      templates" />
    <link rel="http://identifiers.emc.com/linkrel/acls"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/acls" />
    <link rel="http://identifiers.emc.com/linkrel/batches"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/batches" />
    <link rel="http://identifiers.emc.com/linkrel/batch-capabilities"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/batch-
      capabilities" />
    <link rel="http://identifiers.emc.com/linkrel/lifecycles"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/
      lifecycles" />
    <link rel="http://identifiers.emc.com/linkrel/snapshots"
      href="http://localhost:8080/dctm-rest/repositories/REPO1/snapshots" />
  >
  <link rel="http://identifiers.emc.com/linkrel/audit-policies"
    href="http://localhost:8080/dctm-rest/repositories/REPO1/audit-
    policies" />
</links>
</repository>

```



#### Example 2-227: JSON response

```
{
  "id": 12,
  "name": "REPO1",
  "description": "",
  "auth-protocol": "",
  "servers": [
    {
      "name": "REPO1",
      "host": "CS73",
      "version": "7.3.0000.0214 Win64.SQLServer",
      "docbroker": "CS73"
    }
  ],
  "domains": [
    "joli-146",
    "Copy of joli-146"
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO1?include-
      domains=true"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/cabinets",

```

```
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/cabinets"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/checked-out-objects",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/checked-
out-objects"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/current-user",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
currentuser"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/current-user-preferences",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/
currentuser-preferences"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/users",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/users"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/groups",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/groups"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/formats",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/format"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/network-locations",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/network-
locations"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/relations",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/relations"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/relation-types",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/relation-
types"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/types",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/types"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/aspect-types",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO1/aspect-
types"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/dql",
        "hreftemplate": "http://localhost:8080/dctm-rest/repositories/REPO1{?
dql}"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/search",
        "hreftemplate": "http://localhost:8080/dctm-rest/repositories/REPO1/
search{?
collections,facet,include-total,inline,items-per-
```

```

page,locations,
          object-type,page,q,sort,timezone,view}"
},
{
  "rel": "http://identifiers.emc.com/linkrel/saved-searches",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/saved-
searches"
},
{
  "rel": "http://identifiers.emc.com/linkrel/search-templates",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/search-
templates"
},
{
  "rel": "http://identifiers.emc.com/linkrel/acls",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/acls"
},
{
  "rel": "http://identifiers.emc.com/linkrel/batches",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/batches"
},
{
  "rel": "http://identifiers.emc.com/linkrel/batch-capabilities",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/batch-
capabilities"
},
{
  "rel": "http://identifiers.emc.com/linkrel/lifecycles",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/lifecycles"
},
{
  "rel": "http://identifiers.emc.com/linkrel/snapshots",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/snapshots"
},
{
  "rel": "http://identifiers.emc.com/linkrel/audit-policies",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO1/audit-
policies"
}
]
}
}

```



#### Example 2-228: Request and response

##### *Request*

```

GET http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/acme01
Accept:application/vnd.emc.documentum+json

```

##### *Response*

```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "id":1234,
  "name":"acme01",
  "servers":[

```

```
{  
    "name": "acme01",  
    "host": "CS70RC2_Main",  
    "version": "7.0.0000.0501 Win64.SQLServer",  
    "docbroker": "CS70RC2_Main"  
}  
,  
"links": [  
    {  
        "rel": "self",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/users",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/users"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/current-user",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/  
            currentuser"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/groups",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/groups"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/cabinets",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/cabinets"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/formats",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/format"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/network-locations",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/  
            network-locations"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/relations",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/relations"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/relation-types",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/  
            relation-types"  
    },  
    {  
        "rel": "http://identifiers.emc.com/linkrel/checked-out-objects",  
        "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/  
acme01/  
            checked-out-objects"  
    },  
]
```

```
{
  "rel": "http://identifiers.emc.com/linkrel/types",
  "href": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/acme01/types"
},
{
  "rel": "http://identifiers.emc.com/linkrel/dql",
  "hreftemplate": "http://core-rs-demo.lss.emc.com:8080/dctm-rest/repositories/acme01{?dql,page,items-per-page}"
}
]
```



## 2.63 Repositories

The `Repositories` resource represents a collection of available repositories (including those that are not running).

### 2.63.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Home Document” on page 333 and About the Diagram on page 13.

### 2.63.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the repositories without the file extension
- Feed Title: List of repositories
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Repository” on page 564
- Entry ID: URI of the repository without the file extension
- Entry Title: Repository name
- Entry Summary: Repository description

### 2.63.3 Link relations

Link relation	Description	Resource reference
self	Link to this repositories resource.	"Repositories" on page 573

### 2.63.4 Operations

#### 2.63.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of the repository resources.

#### 2.63.4.2 Get repositories

Retrieve a list of repositories with each repository's properties embedded in the corresponding entry.

##### 2.63.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- inline
- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.63.4.2.2 Request media types

N/A

##### 2.63.4.2.3 Request headers

- Accept



**Note:** The Authorization header is not needed when a client tries to get this resource.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.63.4.2.4 Request body

N/A

#### 2.63.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.63.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json (For compatible viewing)

#### 2.63.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.63.4.2.8 Response body

XML or JSON representation of the repository.

## 2.64 Saved Search (es)

### 2.64.1 Saved Search

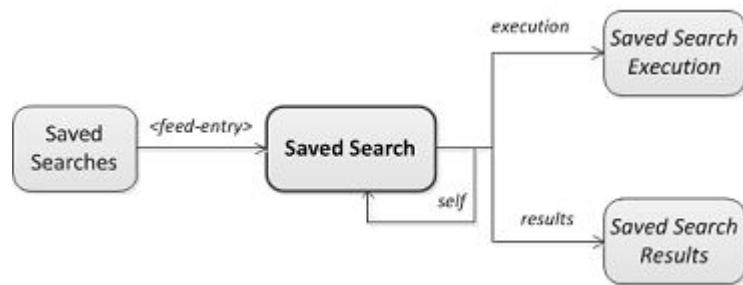
You can define search criteria and save the search. After your search has been saved, you can load, edit the search criteria, or execute the saved search and view the saved results of the search.

### 2.64.1.1 Feed

No feed support.

### 2.64.1.2 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* [About the Diagram on page 13](#).

For more information, see “Saved Searches” on page 586.

### 2.64.1.3 Link relations

Link relation	Description	Resource reference
self	Save the search	<a href="#">“Saved Search” on page 575</a>
edit	Edit the saved search	<a href="#">“Saved Search” on page 575</a>
delete [1]	Delete the saved search	<a href="#">“Saved Search” on page 575</a>
search-execution [1]	Execute the saved search	<a href="#">“Saved Search Execution” on page 596</a>
saved-search-results [1]	View or work with the saved search results	<a href="#">“Saved Search Results” on page 603</a>
as-search-template [1]	Save the saved search as a template	<a href="#">“Saved Search” on page 575</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
`http://identifiers.emc.com/linkrel/`.

## 2.64.1.4 Operations

### 2.64.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves saved search
POST	Updates the saved search
DELETE	Removes the saved search

### 2.64.1.4.2 Get a saved search

Gets a saved search from a repository.

#### 2.64.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

#### 2.64.1.4.2.2 Request media types

N/A

#### 2.64.1.4.2.3 Request headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.64.1.4.2.4 Request body

N/A

#### 2.64.1.4.2.5 Response headers

N/A

#### 2.64.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.64.1.4.2.7 Response status

200 - Retrieved successfully

400, 401, 403, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.64.1.4.2.8 Response body

##### Example 2-229: XML response

```
<saved-search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_smart_list"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
    dm_smart_list.xml">
  <properties>
    <object_name>search iig</object_name>
    <title>this is saved search</title>
    <creation-date>2016-03-10T01:38:44.000+00:00</creation-date>
    <r_modify_date>2016-03-10T01:38:44.000+00:00</r_modify_date>
    <r_modifier>Administrator</r_modifier>
    <owner_name>Administrator</owner_name>
    <r_is_public>true</r_is_public>
    <selected_sources>
      <item>REPO</item>
    </selected_sources>
    <has_results>false</has_results>
    <results_count>-1</results_count>
    <query_type>query_builder</query_type>
    <r_object_id>080000018000157f</r_object_id>
  </properties>
  <query-document>&lt;?xml version='1.0' encoding='UTF-8'?&gt;&lt;search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
    versions="false"
    include-hidden-objects="false"&gt;&lt;repository
    &gt;REPO&lt;/repository&gt;&lt;repositories&gt;&lt;types&gt;&lt;type
    &gt;dm_document&lt;/type&gt;&lt;/types&gt;
    &lt;expression-set operator="AND"&gt;&lt;expressions&gt;
    &lt;fulltext fuzzy="true"&gt;iig&lt;/fulltext&gt;&lt;/expressions&gt;
    &lt;/expression-set&gt;&lt;columns&gt;&lt;column&gt;r_object_type
    &lt;/column&gt;&lt;column&gt;object_name&lt;/column&gt;
    &lt;column&gt;summary&lt;/column&gt;&lt;/columns&gt;&lt;/search&gt;
  &/query-document>
  <links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/
    saved-searches/080000018000157f.xml"/>
    <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/
    saved-searches/080000018000157f.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
    href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
    searches/
    080000018000157f.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/saved-search-results"
    href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
```

```

searches/
    080000018000157f/results.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/search-execution"
        href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
        080000018000157f/execution.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/as-search-template"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
            search-templates.xml"/>
</links>
</saved-search>
```



### Example 2-230: JSON response

```
{
    "name": "saved-search",
    "type": "dm_smart_list",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_smart_list",
    "properties": {
        "object_name": "search iig",
        "title": "this is saved search",
        "creation-date": "2016-03-10T01:38:44.000+00:00",
        "r_modify_date": "2016-03-10T01:38:44.000+00:00",
        "r_modifier": "Administrator",
        "owner_name": "Administrator",
        "r_is_public": true,
        "selected_sources": [
            "REPO"
        ],
        "has_results": false,
        "results_count": -1,
        "query_type": "query_builder",
        "r_object_id": "080000018000157f"
    },
    "query-document": "{\"all-versions\":false, \"include-hidden-objects\"
\"false\",
    \"repositories\":[\"REPO\"], \"types\":[\"dm_document\"], \"expression-set\"
\"expression-type\":"expression-set\", \"operator\":\"AND\",
    \"expressions\":
        [{\"expression-type\":\"fulltext\", \"value\":\"iig\", \"fuzzy\"
\"true\"}],
        \"columns\":[\"r_object_type\", \"object_name\", \"summary\"],
    \"links\": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
                080000018000157f"
        },
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
                080000018000157f"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
```

```

    searches/
        080000018000157f"
    },
{
    "rel": "http://identifiers.emc.com/linkrel/saved-search-results",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
        080000018000157f/results"
},
{
    "rel": "http://identifiers.emc.com/linkrel/search-execution",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
        080000018000157f/execution"
},
{
    "rel": "http://identifiers.emc.com/linkrel/as-search-template",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-
templates"
}
]
}

```



#### 2.64.1.4.3 Update a saved search

You can use this resource to update a saved search. You can update the `object_name`, `title`, and `r_is_public` attributes. You can also update a query-document that contains an AQL. However, the AQL must be escaped.

For more information about the Abstract Query Language, see [AQL](#).

##### 2.64.1.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	query-locale	A locale used to evaluate the query and parse words into a language dependent token	string	N/A	Only supported with <i>xPlore</i> . The default value is <i>en</i> .

#### 2.64.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.64.1.4.3.3 Request headers

- Authorization, Accept, Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.64.1.4.3.4 Request body

Here is a code sample that shows you a saved search, which has properties and an embedded query document in AQL. As you can see, the AQL statements in the <query-document> element must all be escaped as shown below.

##### Example 2-231: XML request

```

<saved-search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <properties>
        <object_name>dave public</object_name>
        <title>this is saved search</title>
        <r_is_public>true</r_is_public>
    </properties>
    <query-document>
        &lt;search xmlns="http://identifiers.emc.com/vocab/documentum"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;
            &lt;types&gt;
                &lt;type&gt;dm_sysobject&lt;/type&gt;
            &lt;/types&gt;
            &lt;expression-set&gt;
                &lt;expressions&gt;
                    &lt;property-list name="owner_name" operator="IN"&gt;
                        &lt;values&gt;
                            &lt;value&gt;dmadmin&lt;/value&gt;
                        &lt;/values&gt;
                    &lt;/property-list&gt;
                &lt;/expressions&gt;
            &lt;/expression-set&gt;
            &lt;facet-definitions&gt;
                &lt;facet-definition id="id-1"&gt;
                    &lt;attributes&gt;
                        &lt;attribute&gt;r_object_type&lt;/attribute&gt;
                    &lt;/attributes&gt;
                &lt;/facet-definition&gt;
            &lt;/facet-definitions&gt;
        &lt;/search&gt;
    </query-document>
</saved-search>

```



▶ **Example 2-232: JSON request**

```
{  
    "properties": {  
        "object_name": "dave public",  
        "title": "this is saved search",  
        "r_is_public": true,  
    },  
    "query-document": "  
    {  
        \"types\": [\"dm_document\"],  
        \"expression-set\": {  
            \"expression-type\": \"expression-set\",  
            \"expressions\": [  
                {  
                    \"expression-type\": \"property\",  
                    \"name\": \"myprop\",  
                    \"operator\": \"CONTAINS\",  
                    \"value\": \"myvalue\"  
                }  
            ]  
        }  
    }  
}
```



#### 2.64.1.4.3.5 Response status

- 200 - OK
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.64.1.4.3.6 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.64.1.4.3.7 Response body

For more information, see [“Response headers” on page 577](#).

▶ **Example 2-233: XML request**

```
<saved-search  
    xmlns="http://identifiers.emc.com/vocab/documentum"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <properties>  
        <object_name>public search</object_name>  
        <title>this is saved search</title>  
        <r_is_public>true</r_is_public>  
    </properties>  
    <query-document>  
        &lt;?xml version='1.0' encoding='UTF-8'?&gt;&lt;search
```

```

        xmlns="http://identifiers.emc.com/vocab/documentum"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
versions="false"
            include-hidden-
objects="false">&lt;repositories>&lt;repository>REPO
&lt;/repository>&lt;/repositories>&lt;types>&lt;type>dm_document&lt;/type>
&lt;/types>&lt;expression-set
operator="AND"&gt;&lt;expressions>&lt;
    fulltext fuzzy="true"&gt;iig&lt;/fulltext&gt;&lt;/
expressions>&lt;/expression-set&gt;
    &lt;columns>&lt;column>r_object_type&lt;/
column&gt;&lt;column>object_name
    &lt;/column&gt;&lt;column>summary&lt;/column&gt;&lt;/
columns>&lt;/search&gt;
    </query-document>
</saved-search>

```



#### ➤ Example 2-234: XML response

```

<saved-search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="dm_smart_list"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_smart_list">
<properties>
    <object_name>public search</object_name>
    <title>this is saved search</title>
    <creation-date>2016-02-03T01:47:42.000+00:00</creation-date>
    <r_modify_date>2016-02-03T01:50:01.000+00:00</r_modify_date>
    <r_modifier>dmadmin</r_modifier>
    <owner_name>dmadmin</owner_name>
    <r_is_public>true</r_is_public>
    <selected_sources>
        <item>REPO</item>
    </selected_sources>
    <has_results>false</has_results>
    <results_count>-1</results_count>
    <query_type>query_builder</query_type>
    <r_object_id>0800000180010d42</r_object_id>
</properties>
<query-document>
    &lt;?xml version='1.0' encoding='UTF-8'?&gt;&lt;search
        xmlns="http://identifiers.emc.com/vocab/documentum"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
versions="false"
            include-hidden-
objects="false">&lt;repositories>&lt;repository>REPO&lt;/repository>&lt;/
repositories>&lt;types>&lt;type>dm_document
    &lt;/type&gt;&lt;types>&lt;expression-set
operator="AND"&gt;&lt;expressions>&lt;
        &lt;fulltext fuzzy="true"&gt;iig&lt;/fulltext&gt;&lt;/
expressions>&lt;/expression-
set&gt;&lt;columns>&lt;column>r_object_type&lt;/
column&gt;
    &lt;column>object_name&lt;/column&gt;&lt;column>summary&lt;/
column&gt;
```

```

        &lt;/columns&gt;&lt;/search&gt;
    </query-document>
    <links>
        <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
REPO/
            saved-searches/0800000180010d42"/>
        <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/
REPO/
            saved-searches/0800000180010d42"/>
        <link rel="http://identifiers.emc.com/linkrel/delete"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
            saved-searches/0800000180010d42"/>
        <link rel="http://identifiers.emc.com/linkrel/saved-search-results"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
            saved-searches/0800000180010d42/results"/>
        <link rel="http://identifiers.emc.com/linkrel/search-execution"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
            saved-searches/0800000180010d42/execution"/>
        <link rel="http://identifiers.emc.com/linkrel/as-search-template"
            href="http://localhost:8080/dctm-rest/repositories/REPO/search-
templates"/>
    </links>
</saved-search>
```



#### Example 2-235: JSON request

```

POST http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
0800000180010d42
Content-Type: application/vnd.emc.documentum+json

{
    "properties": {
        "object_name": "public search",
        "title": "this is saved search",
        "r_is_public": true
    },
    "query-document": "{\"all-versions\":false,\"include-hidden-objects
\":false,\\
        \"repositories\":[\"REPO\"],\"types\":[\"dm_document\"],
        \"expression-set\":{\"expression-type\":\"expression-set\",\"operator\":\"AND
\",
        \"expressions\":[{\"expression-type\":\"fulltext\",\"value\":\"iig\",
        \"fuzzy\":true}]}},\"columns\":[\"r_object_type\", \"object_name\", \"summary
\"]"
}
```



#### Example 2-236: JSON response

```
{
    "name": "saved-search",
    "type": "dm_smart_list",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_smart_list",
    "properties": {
        "object_name": "public search",
        "title": "this is saved search",
```

```

    "creation-date": "2016-02-03T01:47:42.000+00:00",
    "r_modify_date": "2016-02-03T01:48:53.000+00:00",
    "r_modifier": "dmadmin",
    "owner_name": "dmadmin",
    "r_is_public": true,
    "selected_sources": [
        "REPO"
    ],
    "has_results": false,
    "results_count": -1,
    "query_type": "query_builder",
    "r_object_id": "0800000180010d42"
},
"query-document": "{\"all-versions\":false,\"include-hidden-objects\":[],\"repositories\":[\"REPO\"],\"types\":[\"dm_document\"],\"expression-set\":{\"expression-type\":\"expression-set\"},\"operator\":\"AND\"},\"expressions\":[{{\"expression-type\":\"fulltext\",\"value\":\"iig\"},\"fuzzy\":true}}],\"columns\":[\"r_object_type\", \"object_name\"],\"summary\"},\"links\": [
{
    \"rel\": \"self\",
    \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/0800000180010d42\"
},
{
    \"rel\": \"edit\",
    \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/0800000180010d42\"
},
{
    \"rel\": \"http://identifiers.emc.com/linkrel/delete\",
    \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/0800000180010d42\"
},
{
    \"rel\": \"http://identifiers.emc.com/linkrel/saved-search-results\",
    \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/0800000180010d42/results\"
},
{
    \"rel\": \"http://identifiers.emc.com/linkrel/search-execution\",
    \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/0800000180010d42/execution\"
},
{
    \"rel\": \"http://identifiers.emc.com/linkrel/as-search-template\",
    \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/search-templates\"
}
]
}

```



#### **2.64.1.4.4 Delete a saved search**

You can use this resource to delete a saved search.

##### **2.64.1.4.4.1 HTTP Request method and query parameters**

DELETE

##### **2.64.1.4.4.2 Request headers**

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### **2.64.1.4.4.3 Request body**

N/A

##### **2.64.1.4.4.4 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

## **2.64.2 Saved Searches**

This resource retrieves saved searches that are visible to the logged in user. The visible searches include all of the user's saved searches and all shared searches that are available to the user.

### **2.64.2.1 Feed**

Is this resource a feed? Yes.

- Feed ID: <Feed URI of the Saved Searches resource without file extension>
- Feed Title: Saved Searches
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry ID: <Entry URI of the Saved Search>
- Entry Title: Saved Search name
- Entry Updated: Modified date

### 2.64.2.2 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.64.2.3 Link relations

Link relation	Description	Resource reference
self	Saved Searches resource	“Saved Searches” on page 586
first, last, next, previous	Pagination links	Used for navigation when more than one page of search results are available

### 2.64.2.4 Operations

#### 2.64.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the visible saved searches
POST	Creates a saved search

#### 2.64.2.4.2 Get a saved searches collection

Gets the saved searches collection from the repository.

##### 2.64.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• inline</li> <li>• page</li> <li>• items-per-page</li> <li>• total</li> <li>• sort</li> <li>• filter</li> <li>• links</li> </ul>	For more information, see Appendix B, REST Common Definition - URI Request Query Parameters on page 1047.			



### Notes

- The Saved Searches resource ignores the `view` parameter.
- In the feed, the `query-document` is not shown even when `inline=true` has been set.

#### 2.64.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.64.2.4.2.3 Request body

N/A

#### 2.64.2.4.2.4 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.64.2.4.2.5 Response headers

- Content-Type
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.64.2.4.2.6 Response media type

- `application/atom+xml`
- `application/vnd.emc.documentum+json`
- `application/xml`
- `application/json`

#### 2.64.2.4.2.7 Example

► **Example 2-237: XML request**

```
GET http://localhost/dctm-rest/repositories/REPO/saved-searches?items-per-page=1
Accept: application/atom+xml
```



► **Example 2-238: XML response**

```
Status Code: 200 OK
Content-Type: application/atom+xml; charset=UTF-8

<?xml version="1.0" encoding="utf-8"?>
<feed
    xmlns="http://www.w3.org/2005/Atom"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/saved-searches</id>
    <title>Saved Searches</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2015-07-30T02:56:38.870+00:00</updated>
    <dm:page
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
    </dm:page>
    <dm:items-per-page
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">100
    </dm:items-per-page>
    <dm:total
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
    </dm:total>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches.xml?include-total=true&inline=false&view=has_results&sort=object_name%20asc&filter=starts-with(object_name,%27dave%27)"/>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/08024c518000472c</id>
        <title>dave public</title>
        <author>
            <name>dave</name>
        </author>
        <summary>this is saved search</summary>
        <updated>2015-07-30T02:41:25.000+00:00</updated>
        <published>2015-07-30T02:41:25.000+00:00</published>
        <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/08024c518000472c.xml"/>
        <content type="application/xml" src="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/08024c518000472c.xml"/>
    </entry>
</feed>
```



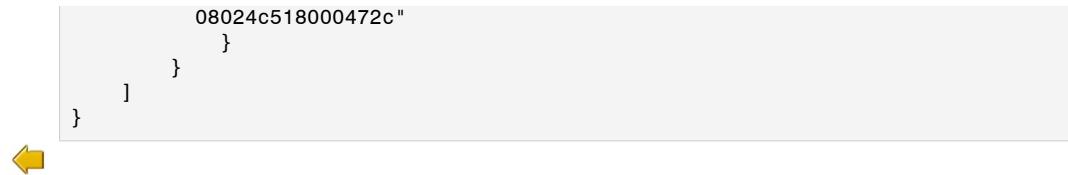
 **Example 2-239: JSON request**

```
GET http://localhost/dctm-rest/repositories/REPO/saved-searches?items-per-
page=1
Accept: application/vnd.emc.documentum+json
```

 **Example 2-240: JSON response**

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches",
    "title": "Saved Searches",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2015-07-30T02:57:34.222+00:00",
    "page": 1,
    "items-per-page": 100,
    "total": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches?
                include-total=true&inline=false&view=has_results&
                sort=object_name%20asc&filter=starts-with(object_name,%27dave%27)"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
                08024c518000472c",
            "title": "dave public",
            "author": [
                {
                    "name": "dave"
                }
            ],
            "summary": "this is saved search",
            "updated": "2015-07-30T02:41:25.000+00:00",
            "published": "2015-07-30T02:41:25.000+00:00",
            "links": [
                {
                    "rel": "edit",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                        saved-searches/
                            08024c518000472c"
                }
            ],
            "content": {
                "type": "application/vnd.emc.documentum+json",
                "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                    saved-searches/"
            }
        }
    ]
}
```



#### 2.64.2.4.3 Create a saved search

For a saved search, there are four basic attributes: name, r\_object\_name, title, r\_is\_public, and query-document. All the other attributes are populated by the system. The name and query-document parameters are mandatory to create a saved search.

##### 2.64.2.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	query-locale	A locale used to evaluate the query and parse words into a language dependent token	string	N/A	Only supported with <i>xPlore</i> . The default value is <i>en</i> .

##### 2.64.2.4.3.2 Request media type

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.64.2.4.3.3 Request headers

- Authorization
- Accept
- For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.64.2.4.3.4 Request body

Here is a code sample that shows you a saved search, which has properties and an embedded query document in AQL. The AQL statements in the <query-document> element must be escaped as shown below.

For more information about the Abstract Query Language, see [AQL](#).

##### ➡ Example 2-241: XML request

```
<saved-search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <properties>
        <object_name>dave public</object_name>
        <title>this is saved search</title>
        <r_is_public>true</r_is_public>
    </properties>
    <query-document>

        &lt;search xmlns="http://identifiers.emc.com/vocab/documentum"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;
            &lt;types&gt;
                &lt;type&gt;dm_sysobject&lt;/type&gt;
            &lt;/types&gt;
            &lt;expression-set&gt;
                &lt;expressions&gt;
                    &lt;property-list name="owner_name" operator="IN"&gt;
                        &lt;values&gt;
                            &lt;value&gt;dmadmin&lt;/value&gt;
                        &lt;/values&gt;
                    &lt;/property-list&gt;
                &lt;/expressions&gt;
            &lt;/expression-set&gt;
            &lt;facet-definitions&gt;
                &lt;facet-definition id="id-1"&gt;
                    &lt;attributes&gt;
                        &lt;attribute&gt;r_object_type&lt;/attribute&gt;
                    &lt;/attributes&gt;
                &lt;/facet-definition&gt;
            &lt;/facet-definitions&gt;
        &lt;/search&gt;

    </query-document>
</saved-search>
```



##### ➡ Example 2-242: JSON request

```
{
    "properties": {
        "object_name": "dmadmin private",
        "title": "this is a facet search",
        "r_is_public": false
    },
    "query-document": "
    {
        \"types\": [\"dm_document\"],
        \"expression-set\": {
```

```

        "expression-type": "expression-set",
        "expressions": [
            {
                "expression-type": "property",
                "name": "myprop",
                "operator": "CONTAINS",
                "value": "myvalue"
            }
        ]
    }
}

```



#### 2.64.2.4.3.5 Response headers

- Location (The URI of the newly created search)
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.64.2.4.3.6 Response media type

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.64.2.4.3.7 Response status

- 201 Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.64.2.4.3.8 Response body



##### Caution

- In AQL, there is an element named `location` that must contain a valid path location. A search with an invalid location can be successfully created. Attempting to load a search that contains an invalid location will cause OpenText Documentum Content Management (CM) Foundation Java API to throw an exception. When this is the case, the REST layer will use the value `UNREACHABLE LOCATION PATH` for the invalid path and no search results can be returned.

- Location must include source information. If you save to a location without using any source information, the system will add a default source (current repository) as the source information. This behavior is consistent with executing a query document.

### Example 2-243: JSON response

```
{  
    "properties": {  
        "object_name": "dmadmin private",  
        "title": "this is a facet search",  
        "creation-date": "2015-07-30T03:06:59.000+00:00",  
        "r_modify_date": "2015-07-30T03:06:59.000+00:00",  
        "r_modifier": "dave",  
        "owner_name": "dave",  
        "r_is_public": false,  
        "selected_sources": [  
            "REPO"  
        ],  
        "has_results": false,  
        "results_count": -1,  
        "query_type": "query_builder",  
        "r_object_id": "08024c518000501f"  
    },  
    "query-document": "{\"all-versions\":false,\"facet-definitions\":[{\"id\":\"\\\"id-1\\\"",br/>        \"attributes\":[\"r_object_type\"],\"group-by\":\"string\",\"sort\":  
        \"FREQUENCY\",  
        \"properties\":{\"skipEmptyValues\":\"true\"},\"max-values\":8}],  
        \"repositories\":[\"REPO\"],\"types\":[\"dm_sysobject\"],  
        \"expression-set\":{  
            \"expression-type\":\"expression-set\",\"operator\":\"AND\",  
            \"expressions\":[{  
                \"expression-type\":\"property-list\",\"name\":\"owner_name\",  
                \"operator\":\"IN\",\"values\":[\"dmadmin\"],\"repeating  
                \":false}]  
            }  
        },  
        \"links\": [  
        {  
            \"rel\": \"self\",  
            \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-  
            searches/  
            08024c518000501f\"\br/>        },  
        {  
            \"rel\": \"edit\",  
            \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-  
            searches/  
            08024c518000501f\"\br/>        },  
        {  
            \"rel\": \"http://identifiers.emc.com/linkrel/delete\",  
            \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-  
            searches/  
            08024c518000501f\"\br/>        },  
        {  
            \"rel\": \"http://identifiers.emc.com/linkrel/saved-search-results\",  
            \"href\": \"http://localhost:8080/dctm-rest/repositories/REPO/saved-  
            searches/\"
```

```

        "08024c518000501f/results"
    },
{
    "rel": "http://identifiers.emc.com/linkrel/search-execution",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
        08024c518000501f/execution"
},
{
    "rel": "http://identifiers.emc.com/linkrel/as-search-template",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-
templates"
}
]
}

```



#### ► Example 2-244: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<saved-search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <properties>
        <object_name>dave public</object_name>
        <title>this is saved search</title>
        <creation-date>2015-07-30T03:09:01.000+00:00</creation-date>
        <r_modify_date>2015-07-30T03:09:01.000+00:00</r_modify_date>
        <r_modifier>dave</r_modifier>
        <owner_name>dave</owner_name>
        <r_is_public>true</r_is_public>
        <selected_sources>
            <item>REPO</item>
        </selected_sources>
        <has_results>false</has_results>
        <results_count>-1</results_count>
        <query_type>query_builder</query_type>
        <r_object_id>08024c5180005021</r_object_id>
    </properties>
    <query-document>
        &lt;?xml version='1.0' encoding='UTF-8'?&gt;
        &lt;search xmlns="http://identifiers.emc.com/vocab/documentum"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            all-versions="false"&gt;
            &lt;repositories&gt;&lt;repository&gt;REPO
            &lt;/repository&gt;&lt;/repositories&gt;&lt;types&
                ;gt;&lt;type&gt;dm_document&lt;/type&gt;
            &lt;/types&gt;&lt;expression-set operator="AND"&gt;&
                lt;expressions&gt;
            &lt;fulltext fuzzy="true"&gt;rest&lt;/fulltext&gt;&
                lt;fulltext fuzzy="false"&gt;test
            &lt;/fulltext&gt;&lt;/expressions&gt;&
                lt;expression-set&gt;&lt;columns&gt;
            &lt;column&gt;r_object_type&lt;/column&gt;&
                lt;column&gt;object_name&lt;/column&gt;
            &lt;column&gt;summary&lt;/column&gt;&lt;/columns&
                gt;&lt;/search&gt;
        </query-document>
        <links>
            <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
                searches/08024c518000501f"/>
        </links>
    </search>
</saved-search>

```

```

        href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
          08024c518000501f.xml"
      ></link>
      <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
          08024c518000501f.xml"
      ></link>
      <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
          08024c518000501f.xml"
      ></link>
      <link rel="http://identifiers.emc.com/linkrel/saved-search-results"
        href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
          08024c518000501f/results.xml"
      ></link>
      <link rel="http://identifiers.emc.com/linkrel/search-execution"
        href="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
          08024c518000501f/execution.xml"
      ></link>
      <link rel="http://identifiers.emc.com/linkrel/as-search-template"
        href="http://localhost:8080/dctm-rest/repositories/REPO/search-
          templates.xml"
      ></link>
    </links>
  </saved-search>

```



### 2.64.3 Saved Search Execution

The Saved Search Execution resource enables you to run a specified saved search and retrieve real-time results.

#### 2.64.3.1 Resource relationships

The following diagram illustrates how this resource is related with other resources:



*See Also:* “Saved Search” on page 575 and About the Diagram on page 13.

### 2.64.3.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the Saved Search Execution resource
- Feed Title: Saved Search Execution
- Feed Updated: Server's current time
- Supports POST: No
- Entry: Search result
- Entry ID: URI of a result
- Entry Name: Result name
- Entry Updated: r\_modify\_date of a result

### 2.64.3.3 Link relations

Link relation	Description	Resource reference
self	This Saved Search Execution resource.	<a href="#">"Saved Search Execution" on page 596</a>
first, last, next, previous	Pagination links.	Used for navigation when more than one page of search results are available

### 2.64.3.4 Operations

#### 2.64.3.4.1 Supported HTTP methods

Method	Description
GET	Retrieve real-time results of a saved search.

#### 2.64.3.4.2 Execute a saved search

Execute a saved search and retrieve real-time results.

##### 2.64.3.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, inline, include-total page, items-per-page	For more information, see Appendix B, REST Common Definition - <a href="#">URI Request Query Parameters on page 1047</a> .			

HTTP method	Query parameters	Description	Data type	Value range	Default value
	facet-id-constraints	<p>Specifies one or more pairs of facet id and its constraints</p> <ol style="list-style-type: none"> <li>1. id1 is the facet for the r_object_type property</li> <li>2. id2 is the facet for the owner_name property</li> </ol> <p>facet-id-constraints=id1%3Ddm_document, id2%3dAdministrator</p>			
	query-locale	A locale used to evaluate the query and parse words into a language dependent token	string	N/A	<p>Only supported with <i>xPlore</i>. The default value is <i>en</i>.</p>



**Note:** International characters and symbols that are used with this query parameter must be URL encoded with the UTF-8 charset. Otherwise, the result may be incorrect.



### Example 2-245: Facet ID constraints

In this example, the facet query is used against the r\_full\_content\_size attribute and the grouping strategy used is range. There are two ranges specified in the AQL: [0, 100000) and [10000, 20000):

```

<search xmlns="http://identifiers.emc.com/vocab/documentum"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <expression-set>
        <expressions>
            <property-list name="owner_name" operator="IN">
                <values>
                    <value>dmadmin</value>
                </values>
            </property-list>
        </expressions>
    </expression-set>
    <facet-definitions>
        <facet-definition group-by="range">
            <attributes>
                <attribute>r_full_content_size</attribute>
            </attributes>
            <properties>
                <property name="range">0:10000,10000:20000</property>
            </properties>
        </facet-definition>
    </facet-definitions>
</search>
```

The search returns the search results along with the facet results. Here you can see that there are two groups returned for [0, 100000) and [10000, 20000)

respectively. Submitting the original AQL to the URL of each group navigates to the results of that group.

```
<dm:facet
  xmlns:dm="http://identifiers.emc.com/vocab/documentum">
  <dm:facet-id>facet_r_full_content_size</dm:facet-id>
  <dm:facet-label>Full Content Size</dm:facet-label>
  <dm:facet-value>
    <dm:facet-id>facet_r_full_content_size</dm:facet-id>
    <dm:facet-value-id>0:10000</dm:facet-value-id>
    <dm:facet-value-count>36</dm:facet-value-count>
    <dm:facet-value-constraint>0/10000</dm:facet-value-constraint>
    <link rel="search" href="http://localhost:8080/dctm-rest/repositories/
REPO/
  search.xml?facet-id-constraints=facet_r_full_content_size%3D0/10000"/>
</dm:facet-value>
<dm:facet-value>
  <dm:facet-id>facet_r_full_content_size</dm:facet-id>
  <dm:facet-value-id>10000:20000</dm:facet-value-id>
  <dm:facet-value-count>10</dm:facet-value-count>
  <dm:facet-value-constraint>10000/20000</dm:facet-value-constraint>
  <link rel="search" href="http://localhost:8080/dctm-rest/repositories/
REPO/
  search.xml?facet-id-constraints=facet_r_full_content_size
%3D10000/20000"/>
</dm:facet-value>
</dm:facet>
```



#### 2.64.3.4.2.2 Request media types

N/A

#### 2.64.3.4.2.3 Request headers

- Accept
- Authorization

#### 2.64.3.4.2.4 Request body

N/A.

#### 2.64.3.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.64.3.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.64.3.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.64.3.4.2.8 Response body

XML or JSON representation of the real-time results of the saved search.

- The body contains a list of search results.
- The returned result collection only contains those results to which the REST client has access.
- Pagination is supported.

##### ➡ Example 2-246: XML response

```
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/saved-searches
  /08024c5180005288/execution</id>
  <title>Search results</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2015-08-04T07:19:05.885+00:00</updated>
  <dm:page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
  </dm:page>
  <dm:items-per-page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
  </dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/
  REPO/saved-searches/08024c5180005288/execution.xml?items-per-page=1"/>
  <link rel="next" href="http://localhost:8080/dctm-rest/repositories/
  REPO/saved-searches/08024c5180005288/execution.xml?items-per-
  page=1&page=2"/>
  <link rel="first" href="http://localhost:8080/dctm-rest/repositories/REPO/
  saved-searches/08024c5180005288/execution.xml?items-per-page=1&page=1"/>
  <link rel="search" hreftemplate="http://localhost:8080/dctm-rest/
  repositories/REPO/search.xml{?collections,facet,include-total,inline,
  items-per-page,locations,object-type,page,q,sort,timezone,view}" />
  <entry>
    <id>09024c518000488c</id>
```

```

<title>ADFS and Shib 0.81.docx</title>
<author>
    <name>UnitTestUser</name>
</author>
<summary>Federated Access 18 Step 3: ...</summary>
<updated>2015-08-04T06:50:23.000+00:00</updated>
<link rel="edit"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
09024c518000488c.xml"/>
<content type="application/xml" src="http://localhost:8080/dctm-rest/
repositories/REPO
/objects/09024c518000488c.xml"/>
<relevance:score
    xmlns:relevance="http://a9.com/-/opensearch/extensions/relevance/
1.0/">1.0
</relevance:score>
<dm:terms
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    <dm:term>Test</dm:term>
    <dm:term>test</dm:term>
    <dm:term>testing</dm:term>
    <dm:term>rest</dm:term>
    <dm:term>Testing</dm:term>
    <dm:term>tested</dm:term>
</dm:terms>
</entry>
...
<entry>...</entry>
</feed>

```



#### ➡ Example 2-247: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
08024c5180005288/execution",
    "title": "Search results",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2015-08-04T07:17:59.804+00:00",
    "page": 1,
    "items-per-page": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
08024c5180005288/execution?items-per-page=1"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
08024c5180005288/execution?items-per-page=1&page=2"
        },
        {
            "rel": "first",

```

```
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
    },
    {
        "rel": "search",
        "hreftemplate": "http://localhost:8080/dctm-rest/repositories/REPO/
search{?collections,facet,include-total,inline,items-per-
page,locations,
object-type,page,q,sort,timezone,view}"
    }
],
"entries": [
{
    "id": "09024c518000488c",
    "title": "ADFS and Shib 0.81.docx",
    "author": [
        {
            "name": "UnitTestUser"
        }
    ],
    "summary": "Federated Access\t18 Step 3: ...",
    "updated": "2015-08-04T06:50:23.000+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
objects/
09024c518000488c"
        }
    ],
    "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
objects/
09024c518000488c"
    },
    "score": "1.0",
    "terms": [
        "Test",
        "test",
        "testing",
        "rest",
        "Testing",
        "tested"
    ]
}
...
{...}
]
}
```



## 2.64.4 Saved Search Results

The Saved Search Results resource allows you to take the results of a search and save it in Documentum CM Server. Saving search results enables you to retrieve search results without having to re-execute the search.

This resource returns saved results, instead of real-time results.

 **Note:** Facet results cannot be saved

### 2.64.4.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Saved Search](#)” on page 575 and [About the Diagram on page 13](#).

### 2.64.4.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the Saved Search Results resource
- Feed Title: Saved Search Results
- Feed Updated: Time when the results are saved.
- Supports POST: No
- Entry: Search result
- Entry ID: URI of a result
- Entry Name: Result name
- Entry Updated: r\_modify\_date of a result

### 2.64.4.3 Link relations

Link relation	Description	Resource reference
self	This Saved Search Results resource.	<a href="#">"Saved Search Results" on page 603</a>
first, last, next, previous	Pagination links.	Used for navigation when more than one page of search results are available

### 2.64.4.4 Operations

#### 2.64.4.4.1 Supported HTTP methods

Method	Description
PUT	Enable or refresh results of a saved search.
GET	Retrieve results of a saved search.
DELETE	Disable saved results.

#### 2.64.4.4.2 Enable and refresh saved results

By default, a newly created Saved Search resource does not save results. This operation enables any previously disabled saved results. This operation causes the value of the `<has_results>` attribute of a saved search to change from `false` to `true`.

When `<has_results>` is already `true`, which means that results saving is already enabled, this operation can be used to refresh the saved results and prevent users from seeing any out-of-date search results.

##### 2.64.4.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	max-results-count	<p>The maximum number of results to for a saved search</p> <p> <b>Note:</b> When -1 is used as the value for the max-results-count parameter, the Request may require more time to retrieve as many matching objects as possible. The number of matching objects returned is only limited by the Foundation Java API configuration. Using a default value of 1000, which returns 10 pages at once by default, greatly improves performance.</p>	integer	<p>-1 and any non-negative integer.</p> <ul style="list-style-type: none"> <li>• -1: Apply the Foundation Java API configuration</li> <li>• Any non-negative integer: the smaller of max-results-count and Foundation Java API configuration takes effect.</li> </ul>	1000

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.64.4.4.2.2 Request media types

N/A

#### 2.64.4.4.2.3 Request headers

- Accept
- Authorization (optional)

#### 2.64.4.4.2.4 Request body

A PUT request is used to turn on saved results.

 **Example 2-248: XML request - application/vnd.emc.documentum+xml or application/xml media types**

```

<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
    08024c5180004332/saved-results</id>
  <title>Search results</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2015-08-04T02:54:54.399+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:items-per-
    page>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
      saved-searches/08024c5180004332/saved-results.xml?items-per-page=1"/>
    <link rel="next" href="http://localhost:8080/dctm-rest/repositories/REPO/
      saved-searches/08024c5180004332/saved-results.xml?items-per-
      page=1&page=2"/>
    <link rel="first" href="http://localhost:8080/dctm-rest/repositories/REPO/
      saved-searches/08024c5180004332/saved-results.xml?items-per-
      page=1&page=1"/>
    <link rel="search" hreftemplate="http://localhost:8080/dctm-rest/
      repositories/REPO/
      search.xml?{collections,facet,include-total,inline,items-per-
      page,locations,
        object-type,page,q,sort,timezone,view}" />
  <entry>
    <id>09024c5180004344</id>
    <title>ADFS and Shib 0.81.docx</title>
    <summary>Federated Access 18 Step 3: Test AD FS 2.0 as the Identity...
      Step 5:
        Test Shibboleth as the Identity Provider and AD FS 2.0 as...
        following sections. AD FS 2.0 The test deployment that is
        created in the...
          domain controller and the federation server in test
        deployments</summary>
    <updated>2015-08-04T03:00:22.950+00:00</updated>
    <link rel="edit"
      href="http://localhost:8080/dctm-rest/repositories/REPO/
        objects/09024c5180004344.xml"/>
    <content type="application/xml" src="http://localhost:8080/dctm-rest/
      repositories/
        REPO/objects/09024c5180004344.xml"/>
    <relevance:score
      xmlns:relevance="http://a9.com/-/opensearch/extensions/relevance/
      1.0/">1.0
    </relevance:score>
    <dm:terms
      xmlns:dm="http://identifiers.emc.com/vocab/documentum">
      <dm:term>Test</dm:term>
      <dm:term>test</dm:term>
      <dm:term>testing</dm:term>
      <dm:term>rest</dm:term>
      <dm:term>Testing</dm:term>
    
```

```

        <dm:term>tested</dm:term>
    </dm:terms>
</entry>
</feed>

```



**Example 2-249: JSON request - application/vnd.emc.documentum+json or application/json media types**

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
          08024c5180004332/saved-results",
    "title": "Search results",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2015-08-04T02:54:54.399+00:00",
    "page": 1,
    "items-per-page": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
          08024c5180004332/saved-results?items-per-page=1"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
          08024c5180004332/saved-results?items-per-page=1&page=2"
        },
        {
            "rel": "first",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-
searches/
          08024c5180004332/saved-results?items-per-page=1
&page=1"
        },
        {
            "rel": "search",
            "hreftemplate": "http://localhost:8080/dctm-rest/repositories/REPO/
              search{?collections,facet,include-total,inline,items-per-
page,locations,object-type,page,q,sort,timezone,view}"
        }
    ],
    "entries": [
        {
            "id": "09024c5180004344",
            "title": "ADFS and Shib 0.81.docx",
            "author": [
                null
            ],
            "summary": "Federated Access\t18 Step 3: Test AD FS 2.0 as the Identity...
Step 5: Test Shibboleth as the Identity Provider and AD FS 2.0 as... following
sections. AD FS 2.0 The test deployment that is created in the... domain
controller and the federation server in test deployments",
            "updated": "2015-08-04T02:55:22.883+00:00",
        }
    ]
}

```

```
    "links": [
      {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
09024c5180004344"
      }
    ],
    "content": {
      "type": "application/vnd.emc.documentum+json",
      "src": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
09024c5180004344"
    },
    "score": "1.0",
    "terms": [
      "Test",
      "test",
      "testing",
      "rest",
      "Testing"
    ]
  }
}
```



#### 2.64.4.4.2.5 Response headers

- Content-Type

#### 2.64.4.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.64.4.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.64.4.4.2.8 Response body

XML or JSON representation of saved results of the search. See the [response body](#) of the Saved Search Execution resource for representation samples.

- This method sets the `has_results` property of a saved search to `true`.
- The `results_count` represents the count of saved results whether or not `max_results_count` has been reached.

#### 2.64.4.4.3 Get saved results

GET saved results of a search.

##### 2.64.4.4.3.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- `links`
- `inline`
- `include-total`
- `page`
- `items-per-page`

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.64.4.4.3.2 Request media types

N/A

##### 2.64.4.4.3.3 Request headers

- `Accept`
- `Authorization` (optional)

##### 2.64.4.4.3.4 Request body

N/A

#### **2.64.4.4.3.5 Response headers**

- Content-Type

#### **2.64.4.4.3.6 Response media types**

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### **2.64.4.4.3.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.64.4.4.3.8 Response body**

XML or JSON representation of saved results of the search. See the [Response body](#) of the Saved Search Execution for representation samples.

- The body contains a list of search results.
- The returned result collection only contains those that the REST client has access to.
- Pagination is supported.

#### **2.64.4.4 Disable saved results**

Disable saved results. The following properties are updated after this operation:

- The `has_results` property of the saved search resource is set to `false`.
- The `results_count` property of the saved search resource is set to -1.

#### **2.64.4.4.4.1 HTTP Request method and query parameters**

DELETE

**2.64.4.4.4.2 Request media types**

N/A

**2.64.4.4.4.3 Request headers**

- Accept
- Authorization (optional)

**2.64.4.4.4.4 Request body**

N/A

**2.64.4.4.4.5 Response headers**

N/A

**2.64.4.4.4.6 Response media types**

N/A

**2.64.4.4.4.7 Response status**

- 204 - Saved results disabled successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.64.4.4.4.8 Response body**

After you disable saved results, the system returns 204 no content with no content in the response body.

## 2.65 Saved Search Template (s)

### 2.65.1 Saved Search Template

There are two ways to create a search template. You can create a search template using a generalized set of search criteria or you can use the existing criteria of a saved search to create a search template.

A search template allows you to share a search with others users. Users who receive the search template can customize it for their own specific use, and can also give it to other users who can customize it further.

In the search template resource, AQL expressions that contain variables have the boolean attribute `template` set to `true`. Variables are set to a predicate value that can be updated at a later time.

An expression containing variables generally follows this pattern. For example:

```

<sample_expression attr_1=v_1 attr_2=v2 ... template="true">predicate_value</
sample_expression>

<property-list name="r_object_type" operator="IN" repeating="false" template="true">
  <values>
    <value>dm_document</value>
  </values>
</property-list>

```

The following table lists all AQL expressions that support variables:

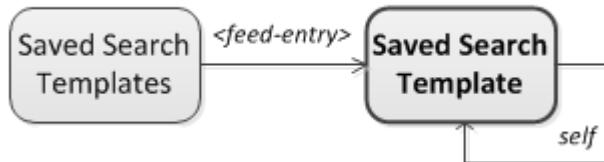
Expression type	Variable	Sample (variable predicates highlighted in bold)
fulltext	Fulltext search criterion	<fulltext fuzzy="false" template="true"> <b>key word</b> </fulltext>
property	Property value   <b>Note:</b> The search operator cannot be IS_NULL or IS_NOT_NULL.	<property name="object_name" operator="CONTAINS" exact-match="false" repeating="false" case-sensitive="false" fuzzy="false" template="true"> <b>myobj</b> </property>
property list	Value list	<property-list name="r_object_type" operator="IN" repeating="false" template="true">   <values>     <value>dm_document</value>     <value>dm_folder</value>   </values> </property-list>
relative date	Relative value and time unit	<relative-date name="r_modify_date" time-unit="MONTH" operator="GREATER_THAN" repeating="false" template="true">-3</relative-date>

Expression type	Variable	Sample (variable predicates highlighted in bold)
property range	From and To values	<pre>&lt;property-range   name="r_modify_date"   operator="BETWEEN"   repeating="false"   template="true"&gt;  &lt;from&gt;2010-10-13T07:33:13.009+00:00&lt;/from&gt;  &lt;to&gt;2014-11-11T01:35:17.089+00:00&lt;/to&gt;  &lt;/property-range&gt;</pre>

A search template differs from a saved search in that a search template may contain variables in its search criteria while a saved search cannot.

### 2.65.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* [About the Diagram on page 13](#).

For more information, see “[Saved Search Templates](#)” on page 621.

### 2.65.1.2 Feed

No feed support.

### 2.65.1.3 Link relations

Link relation	Description	Resource reference
self	This Search Template resource.	<a href="#">"Saved Search Template" on page 611</a>
delete [1]	This Search Template resource.	<a href="#">"Saved Search Template" on page 611</a>
search-execution [1]	Execute the search template	<a href="#">"Search Template Execution" on page 637</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
`http://identifiers.emc.com/linkrel/`.

### 2.65.1.4 Operations

#### 2.65.1.4.1 Supported HTTP methods

Method	Description
GET	Get a search template.
DELETE	Delete a search template.

#### 2.65.1.4.2 Get a Search Template

GET a search template.

##### 2.65.1.4.2.1 HTTP Request method and query parameters

GET

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.65.1.4.2.2 Request media types

N/A

##### 2.65.1.4.2.3 Request headers

- Accept
- Authorization (optional)

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.65.1.4.2.4 Request body**

N/A

**2.65.1.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.65.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

**2.65.1.4.2.7 Response status**

- 200 Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.65.1.4.2.8 Response body**

XML or JSON representation of the search template. Search criteria containing variables are presented in the query-document-template block where all variables have their template attribute set to *true*. Moreover, the external-variables block is appended to query-document-template, which contains a number of *<expression\_type>-var* sections describing all variables in the template.

```
<external-variables>
...
<fulltext-variable>
  <id>/expression[0]</id>
  <expression-type>fulltext</expression-type>
  <data-type>string</data-type>
  <variable-value>key word</variable-value>
</fulltext-variable>
<property-variable>
  <id>/expression[1]/expression[0]</id>
  <expression-type>property</expression-type>
  <property-name>object_name</property-name>
  <data-type>string</data-type>
  <operator>CONTAINS</operator>
  <variable-value>attr1</variable-value>
</property-variable>
...
</external-variables>
```

An expression\_type variable section contains the following elements:

Element	Description
id	xPath-like identifier of the expression where the variable is located. For example, /expression[1]/expression[0] indicates the first child expression of the second expression under the root.
expression-type	Type of the expression where the variable is located.
property-name	Name of the property that is used as a variable. Not valid in fulltext expressions.
data-type	Data type of the property that is used as a variable. Not valid in fulltext expressions.
operator	Operator of the expression where the variable is located. Not valid in fulltext and relative date expressions. The GREATER_THAN operator can be used for relative date expressions.
variable-value	Predicate variable value.

### Example 2-250: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<saved-search-template
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dmc_search_template"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dmc_search_template.xml">
    <properties>
        <object_name>test</object_name>
        <subject>Search template Description for unit tests.</subject>
        <creation-date>2015-09-23T01:45:31.000+00:00</creation-date>
        <r_modify_date>2015-09-23T01:45:31.000+00:00</r_modify_date>
        <owner_name>dadmin</owner_name>
        <r_is_public>true</r_is_public>
        <selected_sources>
            <item>REPO</item>
        </selected_sources>
        <r_object_id>0900000180010d1f</r_object_id>
    </properties>
    <query-document-template>&lt;?xml version='1.0' encoding='UTF-8'?>
        &lt;search
            xmlns="http://identifiers.emc.com/vocab/documentum"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
            versions="false">
            &lt;repositories>&lt;repository>REPO&lt;/repository>
                &lt;/repositories>&lt;types>
                    &lt;type>dm_sysobject&lt;/type>&lt;/types>
                    &lt;expression-set operator="AND">
                        &lt;expressions>&lt;fulltext fuzzy="false" template="true">key word
                            &lt;/fulltext>
                        &lt;expression-set operator="AND">&lt;expressions>
                            &lt;property name="object_name" operator="CONTAINS" exact-
                                match="false"
                                repeating="false"

```

```

        case-sensitive="false" fuzzy="false" template="true">>attr1</;
property>
    &lt;/expressions>
    &lt;/expression-set>&lt;property-range name="r_modify_date"
        operator="BETWEEN"
        repeating="false" template="true">&lt;from>1973-01-26T01:45:31
000+00:00
        &lt;/from>
        &lt;to>2006-01-30T01:12:31 000+00:00&lt;/to>
        &lt;/property-range>&lt;relative-date
            name="r_modify_date" time-unit="MONTH" operator="GREATER_THAN"
            repeating="false"
            template="true">-3&lt;/relative-date>&lt;/expressions>
            &lt;/expression-set>&lt;/search>
        </query-document-template>
<external-variables>
    <fulltext-variable>
        <id>/expression[0]</id>
        <expression-type>fulltext</expression-type>
        <data-type>string</data-type>
        <variable-value>key word</variable-value>
    </fulltext-var>
    <property-variable>
        <id>/expression[1]/expression[0]</id>
        <expression-type>property</expression-type>
        <property-name>object_name</property-name>
        <data-type>string</data-type>
        <operator>CONTAINS</operator>
        <variable-value>attr1</variable-value>
    </property-variable>
    <property-variable>
        <id>/expression[2]/from</id>
        <expression-type>property-range</expression-type>
        <property-name>r_modify_date</property-name>
        <data-type>datetime</data-type>
        <operator>GREATER_EQUAL</operator>
        <variable-value>1973-01-26T01:45:31 000+00:00</variable-value>
    </property-variable>
    <property-variable>
        <id>/expression[2]/to</id>
        <expression-type>property-range</expression-type>
        <property-name>r_modify_date</property-name>
        <data-type>datetime</data-type>
        <operator>LESS_EQUAL</operator>
        <variable-value>2006-01-30T01:12:31 000+00:00</variable-value>
    </property-variable>
    <relative-date-variable>
        <id>/expression[3]</id>
        <expression-type>relative-date</expression-type>
        <property-name>r_modify_date</property-name>
        <data-type>datetime</data-type>
        <variable-value>-3:MONTH</variable-value>
    </relative-date-variable>
</external-variables>
<links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
search-templates/08024c5180004333/execution.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/search-execution"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
search-templates/08024c5180004333.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/

```

```

    repositories/REPO/search-templates/08024c5180004333.xml"/>
  </links>
</saved-search-template>
```



### Example 2-251: JSON response

```
{
  "name": "saved-search-template",
  "type": "dmc_search_template",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
    dmc_search_template",
  "properties": {
    "object_name": "test",
    "subject": "Search template Description for unit tests.",
    "creation-date": "2015-09-23T01:45:31.000+00:00",
    "r_modify_date": "2015-09-23T01:45:31.000+00:00",
    "owner_name": "dmadmin",
    "r_is_public": true,
    "selected_sources": [
      "REPO"
    ],
    "r_object_id": "0900000180010d1f"
  },
  "query-document-template": "{\"all-versions\":false,\"repositories\":
  [\"REPO\"],
  \"types\":[\"dm_sysobject\"],\"expression-set\":{\"expression-type\":
  \"expression-set\",
  \"operator\":\"AND\",\"expressions\":[{\\"expression-type\\\":\\"fulltext\\\",
  \\"value\\\":
  \\"key word\\\",\"fuzzy\":false,\"template\":\"true\"}, {\\"expression-type\\\":
  \"expression-set\",
  \"operator\":\"AND\",\"expressions\":[{\\"expression-type\\\":\\"property\\\",
  \\"name\\\":
  \\"object_name\\\",\"operator\":\"CONTAINS\",\"value\":\"attr1\\\",\"exact-match\\\":
  false,
  \"repeating\":false,\"case-sensitive\":false,\"fuzzy\":false,\"template\\\":
  \"true\\\"]}], {\\"expression-type\\\":\\"property-range\\\",\"name\\\":
  \"r_modify_date\\\",\"operator\\\":
  \\"BETWEEN\\\",\"from\\\":
  \"1973-01-26T01:45:31 000+00:00\",
  \"to\\\":
  \"2006-01-30T01:12:31 000+00:00\",
  \"repeating\":false,\"template\\\":
  \"true\"}, {\\"expression-type\\\":\\"relative-
  date\\\",
  \"name\\\":
  \"r_modify_date\\\",\"value\\\":
  -3,\"time-unit\\\":
  \"MONTH\\\",\"operator\\\":
  \\"GREATER_THAN\\\",\"repeating\":false,\"template\\\":
  \"true\\\"]}]}}",
  "external-variables": [
    {
      "id": "/expression[0]",
      "expression-type": "fulltext",
      "data-type": "string",
      "variable-value": "key word"
    },
    {
      "id": "/expression[1]/expression[0]",
      "expression-type": "property",
      "property-name": "object_name",
      "data-type": "string",
      "operator": "CONTAINS",
      "value": "attr1"
    }
  ]
}
```

```

        "variable-value": "attr1"
    },
    {
        "id": "/expression[2]/from",
        "expression-type": "property-range",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "operator": "GREATER_EQUAL",
        "variable-value": "1973-01-26T01:45:31 000+00:00"
    },
    {
        "id": "/expression[2]/to",
        "expression-type": "property-range",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "operator": "LESS_EQUAL",
        "variable-value": "2006-01-30T01:12:31 000+00:00"
    },
    {
        "id": "/expression[3]",
        "expression-type": "relative-date",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "variable-value": "-3:MONTH"
    }
],
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-
templates/
08024c5180004333"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/search-execution",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-
template/
08024c5180004333/execution"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-
templates/
08024c5180004333"
    }
]
}

```



#### **2.65.1.4.3 Delete a search template**

Delete a search template.

##### **2.65.1.4.3.1 HTTP Request method and query parameters**

DELETE

##### **2.65.1.4.3.2 Request media types**

N/A

##### **2.65.1.4.3.3 Request headers**

- Authorization (optional)

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### **2.65.1.4.3.4 Request body**

N/A

##### **2.65.1.4.3.5 Response headers**

N/A

##### **2.65.1.4.3.6 Response media types**

N/A

##### **2.65.1.4.3.7 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

##### **2.65.1.4.3.8 Response body**

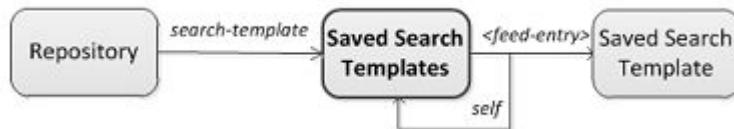
HTTP 204 No Content status upon a successful delete operation. The Response body contains no content.

## 2.65.2 Saved Search Templates

This resource is a collection of search templates. Users can get list of search templates or create a new search template with this resource.

### 2.65.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* [About the Diagram on page 13](#).

For more information, see [“Repository” on page 564](#).

### 2.65.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The URI of the Search Templates resource
- Feed Title: Search templates
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: [“Saved Search Template” on page 611](#)
- Entry ID: Search template URI
- Entry Title: Display name of the search template.
- Entry Updated: `r_modify_date` of the object

### 2.65.2.3 Link relations

Link relation	Description	Resource reference
self	this object	<a href="#">“Saved Search Templates” on page 621</a>
first, last, next, previous	Pagination links	Used for navigation when more than one page of search results are available

## 2.65.2.4 Operations

### 2.65.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a visible search template collection
POST	Creates a new search template

### 2.65.2.4.2 Create a search template

#### 2.65.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	query-locale	A locale used to evaluate the query and parse words into a language dependent token	string	N/A	Only supported with <i>xPlore</i> . The default value is <i>en</i> .

#### 2.65.2.4.2.2 Request headers

- Authorization (optional)
- Accept
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.65.2.4.2.3 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.65.2.4.2.4 Request body

##### ➡ Example 2-252: XML request body

Here is a code sample that shows you a search template that has properties and an embedded query document in AQL. The AQL statements in the <query-document> element must be escaped as shown below.

For more information about Abstract Query Language, see [AQL](#).

```
<search-template
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dmc_search_template"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/
```

```

types/dmc_search_template.xml">
<properties>
    <object_name>wangchen</object_name>
    <subject>Search template Description for unit tests.</subject>
    <r_is_public>true</r_is_public>
</properties>
<query-document-template>
    &lt;search xmlns="http://identifiers.emc.com/vocab/documentum"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
    versions="false"&gt;
        &lt;repositories&gt;
            &lt;repository&gt;REPO&lt;/repository&gt;
        &lt;/repositories&gt;
        &lt;types&gt;
            &lt;type&gt;dm_sysobject&lt;/type&gt;
        &lt;/types&gt;
        &lt;expression-set operator="AND"&gt;
            &lt;expressions&gt;
                &lt;fulltext fuzzy="false" template="true"&gt;key word&lt;/fulltext&gt;
            &lt;expression-set operator="AND"&gt;
                &lt;expressions&gt;
                    &lt;property name="object_name" operator="CONTAINS"
                        exact-match="false" repeating="false" case-sensitive="false"
                        fuzzy="false" template="true"&gt;attr1&lt;/property&gt;
                &lt;/expressions&gt;
            &lt;/expression-set&gt;
            &lt;property-range name="r_modify_date" operator="BETWEEN"
                repeating="false"
                template="true"&gt;
                &lt;from&gt;1973-01-26T03:09:01:000+00:00&lt;/from&gt;
                &lt;to&gt;2015-07-30T03:10:04.000+00:00&lt;/to&gt;
            &lt;/property-range&gt;
            &lt;relative-date name="r_modify_date" time-unit="MONTH"
                operator="GREATER_THAN" repeating="false" template="true"&gt;-3&lt;/
            relative-date&gt;
        &lt;/expressions&gt;
    &lt;/expression-set&gt;
    &lt;/search&gt;
</query-document-template>
</search-template>

```



#### Example 2-253: JSON request body

```
{
  "properties": {
    {
      "object_name": "wangchen",
      "subject": "Search template Description for unit tests.",
      "r_is_public": true
    },
    "query-document-template": "{
        \"all-versions\": false,
        \"repositories\": [
            \"REPO\"
        ],
        \"types\": [
            \"dm_sysobject\"
        ],
        \"expression-set\": "
    
```

```
{  
  \"expression-type\": \"expression-set\",  
  \"operator\": \"AND\",  
  \"expressions\": [  
    {  
      \"expression-type\": \"fulltext\",  
      \"value\": \"key word\",  
      \"fuzzy\": false,  
      \"template\": \"true\"  
    },  
    {  
      \"expression-type\": \"expression-set\",  
      \"operator\": \"AND\",  
      \"expressions\": [  
        {  
          \"expression-type\": \"property\",  
          \"name\": \"object_name\",  
          \"operator\": \"CONTAINS\",  
          \"value\": \"attr1\",  
          \"exact-match\": false,  
          \"repeating\": false,  
          \"case-sensitive\": false,  
          \"fuzzy\": false,  
          \"template\": \"true\"  
        }  
      ]  
    },  
    {  
      \"expression-type\": \"property-range\",  
      \"name\": \"r_modify_date\",  
      \"operator\": \"BETWEEN\",  
      \"from\": \"1973-07-30T03:09:01.000+00:00\",  
      \"to\": \"2015-07-30T035:10:04.000+00:00\",  
      \"repeating\": false,  
      \"template\": \"true\"  
    },  
    {  
      \"expression-type\": \"relative-date\",  
      \"name\": \"r_modify_date\",  
      \"value\": -3,  
      \"time-unit\": \"MONTH\",  
      \"operator\": \"GREATER_THAN\",  
      \"repeating\": false,  
      \"template\": \"true\"  
    }  
  ]  
}
```



#### 2.65.2.4.2.5 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.65.2.4.2.6 Response headers

- Content-Type
- Location: The URI for the newly added search template
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.65.2.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### Example 2-254: XML response

The value of the `<relative-date>` variable has the pattern `number:time-unit` to specify the relative date against the current time.

```
<?xml version="1.0" encoding="UTF-8"?>
<search-template
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dmc_search_template"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
dmc_search_template.xml">
    <properties>
        <object_name>wangchen</object_name>
        <subject>Search template Description for unit tests.</subject>
        <creation-date>2015-09-23T01:45:31.000+00:00</creation-date>
        <r_modify_date>2015-09-23T01:45:31.000+00:00</r_modify_date>
        <owner_name>dmadmin</owner_name>
        <r_is_public>true</r_is_public>
        <selected_sources>
            <item>REPO</item>
        </selected_sources>
        <r_object_id>0900000180010d1f</r_object_id>
    </properties>
    <query-document-template>&lt;?xml version='1.0' encoding='UTF-8'?>
&gt;&lt;search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
versions="false"&gt;&lt;repositories&gt;
    &lt;repository&gt;REPO&lt;/repository&gt;&lt;/
repositories&gt;&lt;types&gt;&lt;type&gt;dm_sysobject&lt;/type&gt;
&lt;/types&gt;&lt;expression-set
```

```
operator="AND">&lt;expressions>&lt;fulltext fuzzy="false"
template="true">&lt;key word&lt;/fulltext&gt;
&lt;expression-set operator="AND">&lt;expressions>&lt;property
name="object_name" operator="CONTAINS"
exact-match="false" repeating="false" case-sensitive="false" fuzzy="false"
template="true">&lt;attr1&lt;/property&gt;&lt;/expressions&gt;&lt;/expression-
set&gt;&lt;property-range name="r_modify_date" operator="BETWEEN"
repeating="false" template="true">&lt;from&gt;01/26/1973 00:00:00&lt;/
from&gt;&lt;to&gt;11/30/2006 00:00:00&lt;/to&gt;&lt;/property-
range&gt;&lt;relative-date name="r_modify_date" time-unit="MONTH"
operator="GREATER_THAN" repeating="false" template="true">-3&lt;/relative-
date&gt;&lt;/expressions&gt;&lt;/expression-set&gt;&lt;/search&gt;;
    </query-document-template>
    <external-variables>
        <fulltext-variable>
            <id>/expression[0]</id>
            <expression-type>fulltext</expression-type>
            <data-type>string</data-type>
            <variable-value>key word</variable-value>
        </fulltext-variable>
        <property-variable>
            <id>/expression[1]/expression[0]</id>
            <expression-type>property</expression-type>
            <property-name>object_name</property-name>
            <data-type>string</data-type>
            <operator>CONTAINS</operator>
            <variable-value>attr1</variable-value>
        </property-variable>
        <property-variable>
            <id>/expression[2]/from</id>
            <expression-type>property-range</expression-type>
            <property-name>r_modify_date</property-name>
            <data-type>datetime</data-type>
            <operator>GREATER_EQUAL</operator>
            <variable-value>2015-09-23T01:45:31.000+00:00</variable-value>
        </property-variable>
        <property-variable>
            <id>/expression[2]/to</id>
            <expression-type>property-range</expression-type>
            <property-name>r_modify_date</property-name>
            <data-type>datetime</data-type>
            <operator>LESS_EQUAL</operator>
            <variable-value>2015-09-23T01:45:31.000+00:00</variable-value>
        </property-variable>
        <relative-date-variable>
            <id>/expression[3]</id>
            <expression-type>relative-date</expression-type>
            <property-name>r_modify_date</property-name>
            <data-type>datetime</data-type>
            <variable-value>-3:MONTH</variable-value>
        </relative-date-variable>
    </external-variables>
    <links>
        <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
            search-templates/08024c5180004333.xml"/>
        <link rel="delete" href="http://localhost:8080/dctm-rest/repositories/REPO/
            search-templates/08024c5180004333/execution.xml"/>
        <link rel="execution"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
            search-templates/08024c5180004333.xml"/>
    
```

```
</links>
</search-template>
```



### Example 2-255: JSON response

```
{
    "name": "search-template",
    "type": "dmc_search_template",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dmc_search_template",
    "properties": {
        "object_name": "wangchen",
        "subject": "Search template Description for unit tests.",
        "creation-date": "2015-09-23T01:45:31.000+00:00",
        "r_modify_date": "2015-09-23T01:45:31.000+00:00",
        "owner_name": "dmadmin",
        "r_is_public": true,
        "selected_sources": [
            "REPO"
        ],
        "r_object_id": "0900000180010d1f"
    },
    "query-document-template": "{\"all-versions\":false,\"repositories\": [\"REPO\"], \"types\":[\"dm_sysobject\"],\"expression-set\":{\"expression-type\":\"expression-set\", \"operator\":\"AND\", \"expressions\":[{\"expression-type\":\"fulltext\", \"value\":\"key word\", \"fuzzy\":false, \"template\":\"true\"}, {\"expression-type\":\"expression-set\", \"operator\":\"AND\", \"expressions\":[{\"expression-type\":\"property\", \"name\": \"object_name\", \"operator\":\"CONTAINS\", \"value\":\"attr1\", \"exact-match\":false, \"repeating\":false, \"case-sensitive\":false, \"fuzzy\":false, \"template\":\"true\"}], {\"expression-type\":\"property-range\", \"name\": \"r_modify_date\", \"operator\":\"BETWEEN\", \"from\":\"1973-09-23T01:45:31.000+00:00\", \"to\":\"2015-09-23T08:45:31.000+00:00\", \"repeating\":false, \"template\":\"true\"}, {\"expression-type\":\"relative-date\", \"name\": \"r_modify_date\", \"value\":-3, \"time-unit\":\"MONTH\", \"operator\":\"GREATER_THAN\", \"repeating\":false, \"template\":\"true\"}]}],
    "external-variables": [
        {
            "id": "/expression[0]",
            "expression-type": "fulltext",
            "data-type": "string",
            "variable-value": "key word"
        },
        {
            "id": "/expression[1]/expression[0]",
            "expression-type": "property",
            "property-name": "object_name",
            "data-type": "string",
            "operator": "CONTAINS",
            "variable-value": "attr1"
        }
    ]
}
```

```
{
  "id": "/expression[2]/from",
  "expression-type": "property-range",
  "property-name": "r_modify_date",
  "data-type": "datetime",
  "operator": "GREATER_EQUAL",
  "variable-value": "2015-09-23T01:45:31.000+00:00"
},
{
  "id": "/expression[2]/to",
  "expression-type": "property-range",
  "property-name": "r_modify_date",
  "data-type": "datetime",
  "operator": "LESS_EQUAL",
  "variable-value": "2015-09-23T01:45:31.000+00:00"
},
{
  "id": "/expression[3]",
  "expression-type": "relative-date",
  "property-name": "r_modify_date",
  "data-type": "datetime",
  "variable-value": "-3:MONTH"
},
],
"links": [
  {
    "rel": "self",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
      search-templates/08024c5180004333"
  },
  {
    "rel": "delete",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
      search-template/08024c5180004333/execution"
  },
  {
    "rel": "execution",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
      search-templates/08024c5180004333"
  }
]
}
```



#### 2.65.2.4.3 Use a saved search as search template

This operation creates a new search template from an existing saved search. Each call to this method creates a new search template. Any expressions that are in the saved search that can be templates will become templates. When a property expression has the IS\_NULL or IS\_NOT\_NULL operator it does not become a template. This is because there is no value in the property expression that can be used as a variable.

#### 2.65.2.4.3.1 HTTP Request method and query parameters

POST

#### 2.65.2.4.3.2 Request media types

application/vnd.emc.documentum+xml

application/vnd.emc.documentum+json

#### 2.65.2.4.3.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.65.2.4.3.4 Request body

The three attributes of the search template that can be specified by users include the following:

- object\_name
- subject
- r\_is\_public

You can set the object\_name attribute, however, when a value for the object\_name attribute is not specified in the request body, its value is set from the saved search used to create the new template. The values for all of the other attributes are set using the values from the source saved search used to create the new template. The search-reference attribute specifies the URL of the source saved search.

##### ➡ Example 2-256: XML request

```
<search-template
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" search-
reference="http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
0800000580008fd3">
    <properties>
        <object_name>new template</object_name>
        <subject>Search template created from saved search.</subject>
        <r_is_public>true</r_is_public>
    </properties>
</search-template>
```



##### ➡ Example 2-257: JSON request

```
{
    "properties": {
```

```
        "object_name": "new_template",
        "subject": "Search template created from saved search.",
        "r_is_public": true
    },
    "search-reference": "http://localhost:8080/dctm-rest/repositories/REPO/
        saved-searches/0800000580008fd3"
}
```



#### 2.65.2.4.3.5 Response headers

- Content-Type
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).
- Location - The URI of the newly created search template.

#### 2.65.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.65.2.4.3.7 Response status

- 201 Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.65.2.4.3.8 Response body



##### Example 2-258: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<search-template
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dmc_search_template"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types
        /dmc_search_template.xml">
<properties>
    <object_name>wangchen</object_name>
    <subject>Search template Description for unit tests.</subject>
    <r_creation_date>2015-09-23T01:45:31.000+00:00</r_creation_date>
    <r_modify_date>2015-09-23T01:45:31.000+00:00</r_modify_date>
    <owner_name>dadmin</owner_name>
    <r_is_public>true</r_is_public>
    <selected_sources>
        <item>REPO</item>
    </selected_sources>

```

```

<r_object_id>0900000180010d1f</r_object_id>
</properties>
<query-document-template><?xml version='1.0' encoding='UTF-8'?><search
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" all-
    versions="false">
    <repositories><repository>REPO</repository>
    </repositories><types><type>dm_sysobject</type>
    </types><expression-set operator="AND"><expressions>
        <fulltext fuzzy="false" template="true">key word</fulltext>
        <expression-set operator="AND"><expressions><property
            name="object_name" operator="CONTAINS" exact-match="false"
            repeating="false"
            case-sensitive="false" fuzzy="false" template="true">attr1</property>
        </expressions></expression-set><property-range name="r_modify_date"
            operator="BETWEEN" repeating="false" template="true">
            <from>01/26/1973 00:00:00</from><to>11/30/2006 00:00:00
            </to></property-range><relative-date name="r_modify_date"
            time-unit="MONTH" operator="GREATER_THAN" repeating="false"
            template="true">
                -3</relative-date></expressions></expression-set>
    </search>
</query-document-template>
<external-variables>
    <fulltext-variable>
        <id>/expression[0]</id>
        <expression-type>fulltext</expression-type>
        <data-type>string</data-type>
        <variable-value>key word</variable-value>
    </fulltext-variable>
    <property-variable>
        <id>/expression[1]/expression[0]</id>
        <expression-type>property</expression-type>
        <property-name>object_name</property-name>
        <data-type>string</data-type>
        <operator>CONTAINS</operator>
        <variable-value>attr1</variable-value>
    </property-variable>
    <property-variable>
        <id>/expression[2]/from</id>
        <expression-type>property-range</expression-type>
        <property-name>r_modify_date</property-name>
        <data-type>datetime</data-type>
        <operator>GREATER_EQUAL</operator>
        <variable-value>01/26/1973 00:00:00</variable-value>
    </property-variable>
    <property-variable>
        <id>/expression[2]/to</id>
        <expression-type>property-range</expression-type>
        <property-name>r_modify_date</property-name>
        <data-type>datetime</data-type>
        <operator>LESS_EQUAL</operator>
        <variable-value>11/30/2006 00:00:00</variable-value>
    </property-variable>
    <relative-date-variable>
        <id>/expression[3]</id>
        <expression-type>relative-date</expression-type>
        <property-name>r_modify_date</property-name>
        <data-type>datetime</data-type>
        <variable-value>-3:MONTH</variable-value>
    </relative-date-variable>
</external-variables>

```

```

<links>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO
    /search-templates/08024c5180004333.xml"/>
  <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/
    search-templates/08024c5180004333.xml"/>
  <link rel="http://identifiers.emc.com/linkrel/delete"
    href="http://localhost:8080/dctm-rest/repositories/REPO/search-templates/
      08024c5180004333.xml"/>
</links>
</search-template>

```



### Example 2-259: JSON response

```
{
  "name": "search-template",
  "type": "dmc_search_template",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
    dmc_search_template",
  "properties": {
    "object_name": "wangchen",
    "subject": "Search template Description for unit tests.",
    "r_creation_date": "2015-09-23T01:45:31.000+00:00",
    "r_modify_date": "2015-09-23T01:45:31.000+00:00",
    "owner_name": "dmadmin",
    "r_is_public": true,
    "selected_sources": [
      "REPO"
    ],
    "r_object_id": "0900000180010d1f"
  },
  "query-document-template": "{\"all-versions\":false,\"repositories\":
  [\"REPO\"],
  \"types\":[\"dm_sysobject\"],\"expression-set\":{\"name\":\"expression-
  set\",
  \"operator\":\"AND\",\"expressions\":[{\"expression-type\":\"fulltext\",
  \"value\":\"key word\",\"fuzzy\":false,\"template\":\"true\"},
  {\"name\":\"expression-set\",\"operator\":\"AND\",\"expressions\":[
    {\"expression-type\":\"property\",\"name\":\"object_name\",
    \"operator\":\"CONTAINS\",\"value\":\"attr1\",\"exact-match\":false,
    \"repeating\":false,\"case-sensitive\":false,\"fuzzy\":false,
    \"template\":\"true\"}],
  {\"expression-type\":\"property-range\",\"name\":\"r_modify_date\",
  \"operator\":\"BETWEEN\",\"from\":\"01/26/1973 00:00:00\",
  \"to\":\"11/30/2006 00:00:00\",\"repeating\":false,\"template\":\"true
  \"},
  {\"expression-type\":\"relative-date\",\"name\":\"r_modify_date\",
  \"value\":-3,\"time-unit\":\"MONTH\",\"operator\":\"GREATER_THAN\",
  \"repeating\":false,\"template\":\"true\"]}}\",
  "external-variables": [
    {
      "id": "/expression[0]",
      "expression-type": "fulltext",
      "data-type": "string",
      "variable-value": "key word"
    },
    {
      "id": "/expression[1]/expression[0]",
      "expression-type": "property",
      "property-name": "object_name",
    }
  ]
}
```

```
        "data-type": "string",
        "operator": "CONTAINS",
        "variable-value": "attr1"
    },
    {
        "id": "/expression[2]/from",
        "expression-type": "property-range",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "operator": "GREATER_EQUAL",
        "variable-value": "01/26/1973 00:00:00"
    },
    {
        "id": "/expression[2]/to",
        "expression-type": "property-range",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "operator": "LESS_EQUAL",
        "variable-value": "11/30/2006 00:00:00"
    },
    {
        "id": "/expression[3]",
        "expression-type": "relative-date",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "variable-value": "-3:MONTH"
    }
],
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-templates/08024c5180004333"
    },
    {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-template/08024c5180004333"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/search-templates/08024c5180004333"
    }
]
}
```



#### **2.65.2.4.4 Get search templates**

GET a collection of search templates.

##### **2.65.2.4.4.1 HTTP Request method and query parameters**

GET

This method supports the following common query parameters:

- inline
- items-per-page
- page
- sort
- filter
- links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### **2.65.2.4.4.2 Request media types**

N/A

##### **2.65.2.4.4.3 Request headers**

- Accept
- Authorization (optional)

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### **2.65.2.4.4.4 Request body**

N/A

##### **2.65.2.4.4.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.65.2.4.4.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.65.2.4.4.7 Response status

- 200 Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.65.2.4.4.8 Response body

##### Example 2-260: XML response

```

<feed
    xmlns="http://www.w3.org/2005/Atom"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/search-templates
    </id>
    <title>Search Templates</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2015-09-23T02:02:59.525+00:00</updated>
    <dm:page
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
    </dm:page>
    <dm:items-per-page
        xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
    </dm:items-per-page>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/
        search-templates.xml?inline=true&items-per-page=1"/>
    <link rel="next" href="http://localhost:8080/dctm-rest/repositories/REPO/
        search-templates.xml?inline=true&items-per-page=1&page=2"/>
    <link rel="first" href="http://localhost:8080/dctm-rest/repositories/REPO/
        search-templates.xml?inline=true&items-per-page=1&page=1"/>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/search-templates/
            0900000180010d15</id>
        <title>wangchen</title>
        <author>
            <name>dmadmin</name>
        </author>
        <summary>Search template Description for unit tests.</summary>
        <updated>2015-09-23T02:02:59.526+00:00</updated>
        <published>2015-09-23T01:25:57.000+00:00</published>
        <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/
            REPO/
            search-templates/0900000180010d15.xml"/>
        <content>
            <dm:search-template
                xmlns:dm="http://identifiers.emc.com/vocab/documentum">
```

```

<dm:properties>
    <dm:object_name>wangchen</dm:object_name>
    <dm:r_object_id>0900000180010d15</dm:r_object_id>
    <dm:r_is_public>1</dm:r_is_public>
    <dm:r_modify_date>2015-09-23T01:25:57.000+00:00
    </dm:r_modify_date>
    <dm:creation-date>2015-09-23T01:25:57.000+00:00
    </dm:creation-date>
    <dm:owner_name>dmadmin</dm:owner_name>
    <dm:selected_sources>
        <dm:item>REPO</dm:item>
    </dm:selected_sources>
    <dm:subject>Search template Description for unit tests.
    </dm:subject>
</dm:properties>
<dm:links>
    <link rel="self" href="http://localhost:8080/dctm-rest/
        repositories/REPO/search-templates/
0900000180010d15.xml"/>
</dm:links>
</dm:search-template>
</content>
</entry>
</feed>

```



### ➡ Example 2-261: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/search-templates",
    "title": "Search Templates",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2015-09-23T02:02:17.107+00:00",
    "page": 1,
    "items-per-page": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                search-templates?inline=true&items-per-page=1"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                search-templates?inline=true&items-per-page=1&page=2"
        },
        {
            "rel": "first",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                search-templates?inline=true&items-per-page=1&page=1"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                search-templates/0900000180010d15",

```

```

        "title": "wangchen",
        "author": [
            {
                "name": "dmadmin"
            }
        ],
        "summary": "Search template Description for unit tests.",
        "updated": "2015-09-23T02:02:17.108+00:00",
        "published": "2015-09-23T01:25:57.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    search-templates/0900000180010d15"
            }
        ],
        "content": {
            "name": "search-template",
            "properties": {
                "object_name": "wangchen",
                "r_object_id": "0900000180010d15",
                "r_is_public": 1,
                "r_modify_date": "2015-09-23T01:25:57.000+00:00",
                "creation-date": "2015-09-23T01:25:57.000+00:00",
                "owner_name": "dmadmin",
                "selected_sources": [
                    "REPO"
                ],
                "subject": "Search template Description for unit tests."
            },
            "links": [
                {
                    "rel": "self",
                    "href": "http://localhost:8080/dctm-rest/repositories/
                        REPO/
                            search-templates/0900000180010d15"
                }
            ]
        }
    }
}

```



### 2.65.3 Search Template Execution

This resource fills in the variables and allows you to retrieve real time search results from a search template.

### 2.65.3.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* [About the Diagram on page 13](#).

For more information, see “[Saved Search Templates](#)” on page 621.

### 2.65.3.2 Feed

Is this resource a feed? Yes.

### 2.65.3.3 Link relations

Link relation	Description	Resource reference
self	This object resource	This resource
first, last, next, previous	Pagination links	Used for navigation when more than one page of search results are available

### 2.65.3.4 Operations

#### 2.65.3.4.1 Supported HTTP methods

Method	Description
POST	Retrieves the results of an instantiated search template

#### 2.65.3.4.2 Get search results of an instantiated search template

This operation is similar to execute as an AQL query document. The AQL used to execute this operation comes from the Documentum CM Server and user inputs.

#### 2.65.3.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	<ul style="list-style-type: none"> <li>• inline</li> <li>• page</li> <li>• items-per-page</li> <li>• include-total</li> <li>• links</li> </ul>	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			
	<p>facet-id-constraints</p> <p>Contains the key value pairs for each facet id and its constraint.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>• You can use <code>&lt;id1&gt;</code> as the facet id for the <code>r_object_type</code> property</li> <li>• You can use <code>&lt;id2&gt;</code> as the facet id for the <code>owner_name</code> property</li> </ul> <p><code>facet-id-constraints=id1=dm_document,id2=Administrator</code></p>	string	N/A	null	

HTTP method	Query parameters	Description	Data type	Value range	Default value
	query-locale	A locale used to evaluate the query and parse words into a language dependent token.  Only supported with <i>xPlore</i> .	string	N/A	The default value is <i>en</i> .

#### 2.65.3.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.65.3.4.2.3 Request headers

- Authorization
- Accept
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.65.3.4.2.4 Request body

- Only the *<external-variables>* parameter will have an effect when used in the request body. The system will ignore the *<query-document-template>* and *<properties>* parameters are ignored when included in the Request.
- In each variable, *<id>* and *<variable-value>* are mandatory (for JSON, *<variable-type>* is required). Users can skip other elements; but if other elements are used in the Request body, they must be correct.
- The date type should comply with this pattern yyyy-MM-dd'T'HH:mm:ss.SSSZ, and as specified in ISO 8601.

#### Example 2-262: XML request

```

<?xml version="1.0" encoding="UTF-8"?>
<search-template xmlns="http://identifiers.emc.com/vocab/documentum" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="dmc_search_template" definition="http://localhost:8080/dctm-rest/repositories/REPO/types/dmc_search_template.xml">
    <external-variables>
        <fulltext-variable>
            <id>/expression[0]</id>
            <variable-value>key word</variable-value>
        </fulltext-variable>
        <property-list-variable>
            <id>/expression[1]</id>
            <expression-type>property-list</expression-type>
        </property-list-variable>
    </external-variables>

```

```

<property-name>object_name</property-name>
<data-type>string</data-type>
<operator>IN</operator>
<variable-values>
    <item>v1</item>
    <item>v2</item>
</variable-values>
</property-list-variable>
<property-variable>
    <id>/expression[2]/expression[0]</id>
    <expression-type>property</expression-type>
    <property-name>object_name</property-name>
    <data-type>string</data-type>
    <operator>CONTAINS</operator>
    <variable-value>attr1</variable-value>
</property-variable>
<property-variable>
    <id>/expression[3]/from</id>
    <variable-value>2014-10-13T07:33:13.009+00:00</variable-value>
</property-variable>
<property-variable>
    <id>/expression[3]/to</id>
    <expression-type>property-range</expression-type>
    <property-name>r_modify_date</property-name>
    <data-type>datetime</data-type>
    <operator>LESS_EQUAL</operator>
    <variable-value>2014-11-13T07:33:13.009+00:00</variable-value>
</property-variable>
<relative-date-variable>
    <id>/expression[4]</id>
    <expression-type>relative-date</expression-type>
    <property-name>r_modify_date</property-name>
    <data-type>datetime</data-type>
    <variable-value>-3:MONTH</variable-value>
    <operator>GREATER_THAN</operator>
</relative-date-variable>
</external-variables>
</search-template>

```



### Example 2-263: JSON request

```
{
  "external-variables": [
    {
      "variable-type": "fulltext-variable",
      "id": "/expression[0]",
      "variable-value": "key word"
    },
    {
      "variable-type": "property-list-variable",
      "id": "/expression[1]",
      "expression-type": "property-list",
      "property-name": "object_name",
      "data-type": "string",
      "operator": "IN",
      "variable-values": [
        "v1",
        "v2"
      ]
    },
    {
      "variable-type": "property-variable",
      "id": "/expression[2]/expression[0]",
      "variable-value": "attr1"
    }
  ]
}
```

```
        "variable-type": "property-variable",
        "id": "/expression[3]/from",
        "expression-type": "property-range",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "operator": "GREATER_EQUAL",
        "variable-value": "2014-10-13T07:33:13.009+00:00"
    },
    {
        "variable-type": "property-variable",
        "id": "/expression[3]/to",
        "expression-type": "property-range",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "operator": "LESS_EQUAL",
        "variable-value": "2014-11-13T07:33:13.009+00:00"
    },
    {
        "variable-type": "relative-date-variable",
        "id": "/expression[4]",
        "expression-type": "relative-date",
        "property-name": "r_modify_date",
        "data-type": "datetime",
        "variable-value": "3:MONTH",
        "operator": "GREATER_THAN"
    }
]
```



#### 2.65.3.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.65.3.4.2.6 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.65.3.4.2.7 Response status

- 200 - Retrieved results successfully
- 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.65.3.4.2.8 Response body

➡ Example 2-264: XML response

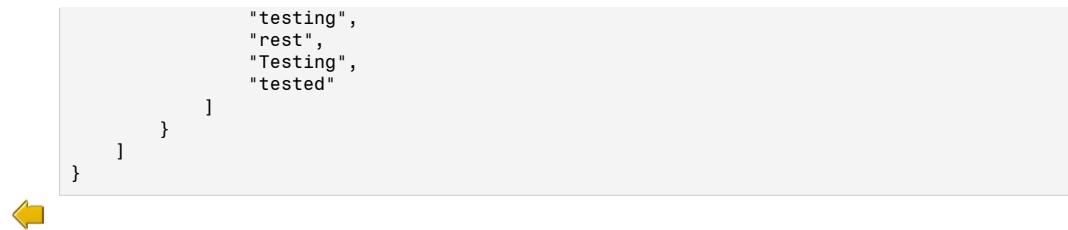
```

<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
  08024c5180005288/
    execution</id>
  <title>Search results</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2015-08-04T07:19:05.885+00:00</updated>
  <dm:page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
  </dm:page>
  <dm:items-per-page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">1
  </dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
  searches/
  08024c5180005288/execution.xml?items-per-page=1"/>
  <link rel="next" href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
  searches/
  08024c5180005288/execution.xml?items-per-page=1&page=2"/>
  <link rel="first" href="http://localhost:8080/dctm-rest/repositories/REPO/saved-
  searches/
  08024c5180005288/execution.xml?items-per-page=1&page=1"/>
  <link rel="search" hreftemplate="http://localhost:8080/dctm-rest/repositories/REPO/
  search.xml{?collections,facet,include-total,inline,items-per-
  page,locations,object-type,
  page,q,sort,timezone,view}"/>
<entry>
  <id>09024c518000488c</id>
  <title>ADFS and Shib 0.81.docx</title>
  <author>
    <name>UnitTestUser</name>
  </author>
  <summary>Federated Access 18 Step 3: Test AD FS 2.0 as the Identity...
    Step 5: Test Shibboleth as the Identity Provider and AD FS 2.0 as...
    following sections. AD FS 2.0 The test deployment that is created in the...
    domain controller and the federation server in test deployments</summary>
  <updated>2015-08-04T06:50:23.000+00:00</updated>
  <link rel="edit" href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
  09024c518000488c.xml"/>
  <content type="application/xml" src="http://localhost:8080/dctm-rest/
  repositories/REPO/
  objects/09024c518000488c.xml"/>
  <relevance:score
    xmlns:relevance="http://a9.com/-/opensearch/extensions/relevance/1.0/">1.0
  </relevance:score>
  <dm:terms
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    <dm:term>Test</dm:term>
    <dm:term>test</dm:term>
    <dm:term>testing</dm:term>
    <dm:term>rest</dm:term>
    <dm:term>Testing</dm:term>
    <dm:term>tested</dm:term>
  </dm:terms>
</entry>
</feed>
```



 **Example 2-265: JSON response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
08024c5180005288",
  "execution",
  "title": "Search results",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2015-08-04T07:17:59.804+00:00",
  "page": 1,
  "items-per-page": 1,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
08024c5180005288/execution?items-per-page=1"
    },
    {
      "rel": "next",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
08024c5180005288/execution?items-per-page=1&page=2"
    },
    {
      "rel": "first",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/saved-searches/
08024c5180005288/execution?items-per-page=1&page=1"
    },
    {
      "rel": "search",
      "hrefTemplate": "http://localhost:8080/dctm-rest/repositories/REPO/
search{?collections,facet,include-total,inline,items-per-page,locations,object-
type,
  page,q,sort,timezone,view}"
    }
  ],
  "entries": [
    {
      "id": "09024c518000488c",
      "title": "ADFS and Shib 0.81.docx",
      "author": [
        {
          "name": "UnitTestUser"
        }
      ],
      "summary": "Federated Access\t18 Step 3: Test AD FS 2.0 as the Identity...
Step 5: Test Shibboleth as the Identity Provider and AD FS 2.0 as...
following sections. AD FS 2.0 The test deployment that is created in the...
domain controller and the federation server in test deployments",
      "updated": "2015-08-04T06:50:23.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
09024c518000488c"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
09024c518000488c"
      },
      "score": "1.0",
      "terms": [
        "Test",
        "test",
        "test",
        "test"
      ]
    }
  ]
}
```



## 2.66 Search

The Search resource enables you to search for objects in a repository according to the criteria that you specify. The search results can be navigated using facets.

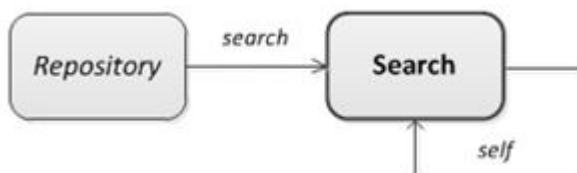
For more information about facet searches, see the Facet Search section in the *OpenText Documentum REST Services Version 7.3 Development Guide*.



**Note:** The Documentum REST Search Service leverages the Foundation Java API to perform search operations on the Documentum CM Server. The property `rest.search.dfc.request.timeout` in the `rest-api-runtime.properties` configuration file specifies a time-out (in milliseconds) for the execution of a Foundation Java API search request. A value of 0 indicates that there is no time-out for search request.

### 2.66.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and [About the Diagram on page 13](#).

### 2.66.2 Feed

Is this resource a feed? Yes.

- Feed ID: URI of the Search resource
- Feed Title: Search
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: A *dm\_sysobject* or a subtype

This is a paged feed, and as such, this feed supports pagination.

## 2.66.3 Link relations

Link relation	Description	Resource reference
self	The URI for Search results sysobjects feed	<a href="#">"Search" on page 645</a>
first, last, next, previous	Pagination links	Used for navigation when more than one page of search results are available
search	The URI template or actual URI for subsequent searches (for example: facet selection)	<a href="#">"Search" on page 645</a>

## 2.66.4 Operations

### 2.66.4.1 Supported HTTP methods

Method	Description
GET	Full-text search using URI parameters.
POST	Full-text search using a query document and URI parameters.

### 2.66.4.2 Full-text search using URI parameters

This operation enables you to perform a full-text search using Simple Search Language along with URI parameters. The operation also supports basic facet navigation.

#### 2.66.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	q	<p>Specifies the search criterion with a full-text expression in Simple Search Language.</p> <p>Parameter q must be encoded because it may contain non-English locale characters.</p> <p> <b>Note:</b> International characters that are used in this query parameter must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.</p>	string	N/A	No search criteria is used for the search.

HTTP method	Query parameters	Description	Data type	Value range	Default value
	locations	<p>Specifies a list of folder paths separated by comma (,) to which the query is scoped.</p> <p>Parameter locations must be encoded because it may contain non-English locale characters.</p> <p> <b>Note:</b> International characters that are used in this query parameter must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.</p>	string	N/A	No location scope constraints are used for the search.
	collections	Specifies a list of collections separated by comma (,), to which the query is scoped. The collections must exist in xPlore.	string	N/A	No collection constraints are used for the search.
	facet	Specifies the property to be used as facet	string	N/A	No property is used for the search.

HTTP method	Query parameters	Description	Data type	Value range	Default value
	facet-value-constraints	<p>Specifies a property constraint expression. For example, when facet is set to r_object_type and facet-value-constraints is set to dm_object, only dm_object instances are returned.</p> <p>You can use boolean operators + for AND and   for OR in a property constraint expression to define more complex constraints. For example, the expression dm_object dm_folder returns both dm_object and dm_folder instances.</p> <p>Parameter facet-value-constraints must be encoded because it may contain non-English locale characters.</p> <p> <b>Note:</b> International characters that are used in this query parameter must be sent with URL encoded by the UTF-8 charset. Otherwise, the result may be incorrect.</p>	string	N/A	No property constraints are used for the search.
	timezone	<p>Indicates the time zone used to compute date facets.</p> <p>Valid time zones include (case-insensitive):</p> <ul style="list-style-type: none"> <li>• Abbreviation, such as GMT and UTC</li> <li>• Full names, such as America/Los_Angeles</li> <li>• Custom time zones, such as GMT -8:00</li> </ul>	string	N/A	GMT
	object-type	Specifies an object type. Only instances of the specified type or its sub types are returned in the search result	string	N/A	dm_SysObject

HTTP method	Query parameters	Description	Data type	Value range	Default value
	query-locale	A locale used to evaluate the query and parse words into a language dependent token	string	N/A	This is only supported with <i>xPlore</i> . The default value is <i>en</i>

#### 2.66.4.2.1.1 URI query parameters

This operation supports the following URI query parameters:

- `include-total`  
The `include-total` parameter in the Search service returns the total number of hit results.
- `inline`
- `items-per-page`
- `page`
- `sort`

The sort order is determined by the data type of the attribute. For example, when `<r_content_size>` is used with an attribute value of 123,3555 to sort search results, the data type is integer and REST sorts the search results by integer.

- `view`

#### Notes

- Pagination parameters (`page`, `items-per-page`, and `include-total`) only take effect on full text search results. As a result, pagination link relations do not appear when you navigate to the result set of a certain facet by using the search link relation in a `facet` section.

A facet is not calculated against the current page, but is instead calculated against the entire result set. For example, if there are 25 results returned for a keyword, the parameter `items-per-page` may have a value of 2. In this case, the Response contains only 2 results, even though the facets were calculated against 25 results.

- The `inline` parameter must be *true*, or the `view` parameter has no effect.
- Setting the `view` parameter to `:default` and to `:all` behave the same. Both settings return the following properties of an object:
  - `r_object_id`
  - `object_name`
  - `r_object_type`

- r\_modify\_date
- r\_modifier
- summary

 **Example 2-266: A GET request**

Here is an example of a GET request to perform a full-text search using URI parameters:

```
http://localhost:8080/dctm-rest/repositories/acme01/search?q=search_criterion  
&facet=property_name&facet-value-constraints=f_v_constraints
```



For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### **2.66.4.2.2 Request media types**

N/A

#### **2.66.4.2.3 Request headers**

- Accept
- Content-Type
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### **2.66.4.2.4 Request body**

N/A

#### **2.66.4.2.5 Response headers**

- Content-Type
- Content-Length

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.66.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/json
- application/xml

#### 2.66.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.66.4.2.8 Response body

XML or JSON representation of the search results with facets.

- The body must contain a list of sysobjects (or subtypes of dm\_SysObject), each of which may or may not contain all attributes of a SysObject, depending on the setting of the view and inline query parameters.
- The returned sysobjects collection only contains objects for which the user has access.
- Pagination is supported.
- Default sort order is the ranking returned by the search engine and it can be overridden with any SysObject attribute.
- Total count can be returned only when it is explicitly specified in the request parameter, along with it the link for last page can be returned.

#### ➡ Example 2-267: XML response - feed

```
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:relevance="http://a9.com/-/opensearch/extensions/relevance/1.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:dm="http://identifiers.emc.com/vocab/documentum">
  ...
  <!--The URI of the GET request sent to the REST server-->
  <link rel="self"
    href="http://localhost:8080/dctm-rest/repositories/acme01/search?
    q=test&facet=r_object_type"/>
  <!--A URI template that lists all available query parameters you can use-->
  <link rel="search"
    hreftemplate="http://localhost:8080/dctm-rest/repositories/acme01/search
    {?q,collections,locations,facet,inline,page,items-per-page,include-total,
    sort,view,facet-value-constraints}/>
  <!--Search results are listed in the following entries-->
  <entry>
    ...
  </entry>
  ...
  ...
```

```

<entry>
  ...
</entry>
<!--Facet information-->
<dm:facet>
  <!--The property used as the facet-->
  <dm:facet-id>r_object_type</dm:facet-id>
  <dm:facet-label>Type</dm:facet-label>
  <!--Data of the results belonging to this fact-->
  <dm:facet-value>
    <dm:facet-value-id>san_biz</dm:facet-value-id>
    <dm:facet-value-count>13</dm:facet-value-count>
    <dm:facet-id>r_object_type</dm:facet-id>
    <dm:facet-value-constraint>san_biz</dm:facet-value-constraint>
    <!--A link that navigate you to the corresponding results of this fact-->
  >
    <link rel="search"
      href="http://localhost:8080/dctm-rest/repositories/acme01/search?
      q=test&facet=r_object_type&fact-vale-constraints=san_biz"/>
  </dm:facet-value>
  <dm:facet-value>
    <dm:facet-value-id>dm_document</dm:facet-value-id>
    <dm:facet-value-count>12</dm:facet-value-count>
    <dm:facet-id>r_object_type</dm:facet-id>
    <dm:facet-value-constraint>dm_document</dm:facet-value-constraint>
    <link rel="search"
      href="http://localhost:8080/dctm-rest/repositories/acme01/search?
      q=test&facet=r_object_type&fact-vale-constraints=dm_document"/>
  </dm:facet-value>
</dm:facet>
</feed>

```



### Example 2-268: JSON response - feed

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/acme01/search",
  "updated": "2014-04-30T14:27:49.643+08:00",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "page": 1,
  "items-per-page": 3,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/acme01/search.json?
      q=test
          &items-per-page=3&inline=false&include_total=true"
    },
    {
      "rel": "next",
      "href": "http://localhost:8080/dctm-rest/repositories/acme01/search.json?
      q=test
          &items-per-page=3&inline=false&include_total=true&page=2"
    },
    {
      "rel": "first",
      "href": "http://localhost:8080/dctm-rest/repositories/acme01/search.json?
      q=test
          &items-per-page=3&inline=false&include_total=true&page=1"
    },
    {
      "rel": "search",
      "hreftemplate": "http://localhost:8080/dctm-rest/repositories/acme01/
          &items-per-page=3&inline=false&include_total=true&q={query}&facet={facet}&constraint={constraint}"
    }
  ]
}
```

```

        search.json{?q,collections,locations,facet,inline,page,items-per-page,
        include-total,sort,view}"
    }
],
"entries": [
{
    "id": "090004d280005117",
    "title": "Native UCF Functional Test Plan.doc",
    "updated": "2013-05-07T15:56:45.000+08:00",
    "summary": "Native UCF Functional Test Plan\t\t09/13/2010 OpenText | Documentum...
Project Native UCF Functional Test Plan Document.../Updated some tests
Functional Test Plan Template Version: 1.2... _Toc408805185 _Toc262725323 _Toc272245663
1.
Test Plan Identifier",
"author": [
{
    "name": "dmadmin"
}
],
"content": {
    "content-type": "application/json",
    "src": ""
},
"links": [
{
    "rel": "self",
    "href": "http://localhost:8080/dctm-rest/repositories/acme01/objects/090004d280005117.json"
}
],
"score": "1.0",
"terms": [
    "Test",
    "Tested",
    "tests",
    "test",
    "testing",
    "Testing",
    "tested"
]
},
{
    "id": "090004d280009f72",
    "title": "TESTED",
    "updated": "2013-09-10T15:33:43.000+08:00",
    "author": [
{
    "name": "dmadmin"
}
],
"content": {
    "content-type": "application/json",
    "src": ""
},
"links": [
{
    "rel": "self",
    "href": "http://localhost:8080/dctm-rest/repositories/acme01/objects/090004d280009f72.json"
}
],
"score": "0.4935832917690277",
"terms": [
    "test"
]
},
{

```

```
        "id": "090004d2800072ae",
        "title": "AAA BACKFLOW TESTING LLC",
        "updated": "2013-09-10T15:25:15.000+08:00",
        "author": [
            {
                "name": "dmadmin"
            }
        ],
        "content": {
            "content-type": "application/json",
            "src": ""
        },
        "links": [
            {
                "rel": "self",
                "href": "http://localhost:8080/dctm-rest/repositories/acme01/objects/090004d2800072ae.json"
            }
        ],
        "score": "0.48992398381233215",
        "terms": [
            "test"
        ]
    }
}
```



#### 2.66.4.3 Full-text search using query document and URI parameters

Retrieves a list of sysobjects or SysObject subtypes that match the search criteria, which means that you can perform a search on metadata.

##### 2.66.4.3.1 HTTP Request method and query parameters

POST

The following URI parameters are supported:

- include-total
- page
- items-per-page
- inline
- facet-id-constraints

Contains the key value pairs for each facet id and its constraint. The id should comply with the facet id in the AQL.

For example:

- You can use <id1> as the facet id for the <r\_object\_type> property
- You can use <id2> as the facet id for the <owner\_name> property

```
facet-id-constraints=id1=dm_document,id2=Administrator
```



**Note:** Using any other search URI parameters not shown above with AQL causes an error message to be shown in the Foundation REST API system.

#### 2.66.4.3.2 Request media types

- application/vnd.emc.documentum+json
- application/vnd.emc.documentum+xml

#### 2.66.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.66.4.3.4 Request body

XML or JSON representation of the query document.

##### ➡ Example 2-269: XML request

```
POST http://localhost:8080/dctm-rest/repositories/acme01/search.xml
Content-Type: application/vnd.emc.documentum

<search>
  <types>
    <type>dm_sysobject</type>
  </types>
  <expression-set>
    <fulltext fuzzy="true">dmadmin</fulltext>
  </expression-set>
  <columns>
    <column>object_name</column>
    <column>r_object_id</column>
  </columns>
  <facets>
    <facet id="r_modify_date">
      <attributes><attribute>r_modify_date</attribute></attributes>
    </facet>
  </facets>
</search>
```



#### 2.66.4.3.5 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.66.4.3.6 Response headers

- Content-Type
- Content-Length

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.66.4.3.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.66.4.3.8 Response body

Same response body as for the GET method: see [“Response body” on page 651](#).

#### ► Example 2-270: XML request

```
POST http://localhost:8080/dctm-rest/repositories/acme01/search
Content-Type: application/vnd.emc.documentum+xml

<search all-versions="true" max-results-for-facets="10"
        xmlns="http://identifiers.emc.com/vocab/documentum"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <types>
        <type>dm_document</type>
    </types>
    <columns>
        <column>r_object_type</column>
        <column>object_name</column>
    </columns>
    <sorts>
        <sort ascii="false" ascending="true" lang="fr">r_content_size</sort>
        <sort ascii="true" ascending="false" lang="zh">object_name</sort>
    </sorts>
    <collections>
        <collection>collection1</collection>
        <collection>collection2</collection>
    </collections>
    <locations>
        <id-location descendent="true">
            <id>0c00000580001914</id>
        </id-location>
        <path-location descendent="false">
            <path>/dmadmin</path>
        </path-location>
    </locations>
</search>
```

```

</locations>
<facet-definitions>
    <facet-definition id="id1" group-by="alpharange" max-values="8">
        <attributes>
            <attribute>r_object_type</attribute>
        </attributes>
        <sort>FREQUENCY</sort>
        <properties>
            <property name="range">a:e, f:z</property>
        </properties>
    </facet-definition>
    <facet-definition id="id2" group-by="relativeDate" max-values="9">
        <attributes>
            <attribute>r_modify_date</attribute>
        </attributes>
        <sort>VALUE_ASCENDING</sort>
    </facet-definition>
</facet-definitions>
<expression-set>
    <expressions>
        <property name="object_name" operator="CONTAINS" exact-match="true"
repeating="true"
            case-sensitive="false" fuzzy="true">rest
        </property>
        <fulltext fuzzy="true">rest</fulltext>
        <fulltext fuzzy="false">test</fulltext>
        <property-range name="r_content_size" operator="BETWEEN" repeating="false">
            <from>0</from>
            <to>10000</to>
        </property-range>
        <property-list name="r_object_type" operator="IN" repeating="false">
            <values>
                <value>dm_document</value>
                <value>dm_folder</value>
            </values>
        </property-list>
        <expression-set>
            <expressions>
                <fulltext fuzzy="false">test</fulltext>
            </expressions>
        </expression-set>
        <relative-date name="r_modify_date"
time-unit="MONTH" operator="GREATER_THAN" >2</relative-date>
    </expressions>
</expression-set>
</search>

```



### Example 2-271: JSON request

```

POST http://localhost:8080/dctm-rest/repositories/acme01/search
Content-Type: application/vnd.emc.documentum+json

{
    "types": ["dm_document"],
    "columns": [
        "r_object_type",
        "object_name"
    ],
    "collections": [
        "collection1",
        "collection2"
    ],
    "locations": [
        {
            "location-type": "id-location",
            "id": "0c00000580001914",

```

```
        "descendent": true
    },
    {
        "location-type": "path-location",
        "path": "/dmadmin",
        "descendent": false
    }
],
"sorts": [
    {
        "property": "r_content_size",
        "ascending": true,
        "lang": "fr",
        "ascii": false
    },
    {
        "property": "object_name",
        "ascending": false,
        "lang": "zh",
        "ascii": true
    }
],
"facet-definitions": [
    {
        "id": "id1",
        "attributes": ["r_object_type"],
        "group-by": "alpharange",
        "sort": "FREQUENCY",
        "properties": {
            "range": "a:e, f:z",
        },
        "max-values": 8
    },
    {
        "id": "id2",
        "attributes": ["r_modify_date"],
        "group-by": "r_modify_date",
        "sort": "VALUE_ASCENDING",
        "max-values": 9
    }
],
"expression-set": {
    "expression-type": "expression-set",
    "expressions": [
        {
            "expression-type": "property",
            "name": "object_name",
            "operator": "CONTAINS",
            "value": "rest",
            "exact-match": true,
            "repeated": true,
            "case-sensitive": false,
            "fuzzy": true
        },
        {
            "expression-type": "fulltext",
            "value": "rest",
            "fuzzy": true
        },
        {
            "expression-type": "fulltext",
            "value": "test",
            "fuzzy": false
        },
        {
            "expression-type": "property-range",
            "name": "r_content_size",
            "operator": "BETWEEN",
            "from": "0",
            "to": "1000"
        }
    ]
}
```

```

        "to": "10000",
        "repeating": false
    },
    {
        "expression-type": "property-list",
        "name": "r_object_type",
        "operator": "IN",
        "values": [
            "dm_document",
            "dm_folder"
        ],
        "repeating": false
    },
    {
        "expression-type": "expression-set",
        "expressions": [
            {
                "expression-type": "fulltext",
                "value": "test",
                "fuzzy": false
            }
        ]
    },
    {
        "expression-type": "relative-date",
        "name": "r_modify_date",
        "value": 2,
        "time-unit": "MONTH",
        "operator": "EQUAL"
    }
]
}

```



#### Example 2-272: Response

```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8
Date: Tue, 19 Feb 2013 10:15:21 GMT
Server: Apache-Coyote/1.1
Transfer-Encoding: chunked

```



#### Example 2-273: JSON response

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/acme01/search",
    "updated": "2014-04-30T14:27:49.643+08:00",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "page": 1,
    "items-per-page": 3,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/0c10304585007313/execution?items-per-page=3&inline=false&include_total=true"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/0c10304585007313/execution?items-per-page=3&inline=false&include_total=true&page=2"
        }
    ]
}

```

```

        "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/
          0c10304585007313/execution?items-per-page=3&inline=false&
          include_total=true&page=2"
    },
{
  "rel": "first",
  "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/
    0c10304585007313/execution?items-per-page=3&inline=false&
    include_total=true&page=1"
}
],
"entries": [
  ...
  ...
]
}

```



#### 2.66.4.4 Search with facets

Faceted searching allows you to view a collection of search results information by applying one or more filters. Faceted searching, also known as Faceted Navigation, uses a classification system to divide information into dimensions, called facets, allowing the classifications to be used in many ways.

REST services supports faceted navigation in two ways:

- URI parameter (facet) in GET request
- Abstract Query Language (AQL) in the POST body

##### 2.66.4.4.1 Facets using query parameters

A URI facet parameter can be used to specify one attribute as the facet. By default, the attributes of SysObject that are facet-enabled depends on the xPlore configuration. The configuration for xPlore must be modified and re-indexed to facet new attributes or add additional facet values to existing attributes.

Attribute	Foundation Java API Data type	Default Grouped By
r_modifier	DF_STRING	string
keywords	DF_STRING	string
r_modify_date	DF_TIME	relativeDate
r_full_content_size	DF_DOUBLE	string
a_application_type	DF_STRING	string
r_object_type	DF_STRING	string
owner_name	DF_STRING	string

The strategy used to group facets depends on the data type of the attribute. Additional grouping strategies can be achieved using AQL language in a POST statement.

For more information, see “[Facet using query document](#)” on page 662.

Another URI parameter called `facet-value-constraints` is used to define an expression that outlines the constraints that are used in the parameter facet. This parameter allows you to navigate a result set under the constraints that your expression defines for the attribute in the parameter facet. Facets used in a feed are returned with the full text search results. You can view the facets in the following response sample:

```
"facets": [
  {
    "facet-id": "facet_r_object_type",
    "facet-value": [
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/search?facet=r_object_type&facet-value-constraints=dm_document",
          "rel": "search"
        },
        "facet-id": "facet_r_object_type",
        "facet-value-constraint": "dm_document",
        "facet-value-count": 116,
        "facet-value-id": "dm_document"
      },
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/search?facet=r_object_type&facet-value-constraints=dm_folder",
          "rel": "search"
        },
        "facet-id": "facet_r_object_type",
        "facet-value-constraint": "dm_folder",
        "facet-value-count": 68,
        "facet-value-id": "dm_folder"
      },
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/search?facet=r_object_type&facet-value-constraints=dm_cabinet",
          "rel": "search"
        },
        "facet-id": "facet_r_object_type",
        "facet-value-constraint": "dm_cabinet",
        "facet-value-count": 25,
        "facet-value-id": "dm_cabinet"
      },
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/search?facet=r_object_type&facet-value-constraints=dm_sysobject",
          "rel": "search"
        },
        "facet-id": "facet_r_object_type",
        "facet-value-constraint": "dm_sysobject",
        "facet-value-count": 12,
        "facet-value-id": "dm_sysobject"
      },
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/search?facet=r_object_type&facet-value-constraints=rest_core",
          "rel": "search"
        },
        "facet-id": "facet_r_object_type",
        "facet-value-constraint": "rest_core",
        "facet-value-count": 1,
        "facet-value-id": "rest_core"
      }
    ]
  }
]
```

```
        "facet-value-constraint": "rest_core",
        "facet-value-count": 8,
        "facet-value-id": "rest_core"
    },
    {
        "link": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/search?
facet=r_object_type&
facet-value-constraints=dm_network_location_map",
            "rel": "search"
        },
        "facet-id": "facet_r_object_type",
        "facet-value-constraint": "dm_network_location_map",
        "facet-value-count": 2,
        "facet-value-id": "dm_network_location_map"
    }
],
"facet-label": "Type"
}
]
```

#### 2.66.4.4.2 Facet using query document

Facet navigation using **AQL** allows you to harness the full power of facets. There are three kinds of facet requests:

- Single facet
- Multiple facet
- Nested facet



**Note:** From an implementation perspective, the facet constraints are defined by the underlying grouping strategy. You can determine facets by the grouping strategies used. The grouping strategies available for a given attribute depends on the data type of that attribute. For example, it is not meaningful to use a day or relativeDate grouping strategy for a string type attribute, such as `object_name`.

*Grouping Strategies*

Strategy	Description	Options (properties)
string	Applies to string values.	<ul style="list-style-type: none"> <li>• <code>nullValueFacet</code> (optional): if this property is set, null values will be counted as a facet. The facet will be named using the value of the property. By default, the null values are ignored when computing facets.</li> <li>• <code>keepDuplicateValues</code> (optional): <code>true/false</code>. Default: <code>false</code>. For a document, attributes with repeating values may have duplicate values. By default, duplicates are removed. If this property is set to <code>true</code>, duplicates entries are counted. Since: xPlore 1.1 P03 and 1.2</li> <li>• <code>caseInsensitive</code> (optional): <code>true/false</code>. Default: <code>false</code>. If this property is set to <code>true</code>, facet computation ignores case. For example, if a results set has a document with a value 'opentext' and another 'OPENTEXT', then two facets will be returned, opentext(1) and OPENTEXT(1). Since: xPlore 1.1 P03 and 1.2</li> </ul>
range	<p>Applies to numeric values.</p> <p>Range grouping requires the use of the <code>&lt;range&gt;</code> property, which defines the ranges to use in grouping.</p> <p>Example:  <code>0:1000,1000:10000,1000  0:</code></p>	<p><code>range</code> (required). This property defines a list of 'buckets' to use for grouping. Each bucket is separated by comma (','). A bucket is defined by a lower and upper bounds separated by colon (:). Open ended bucket can be defined by omitting one of the bounds but keeping the comma separator. Overlapping buckets are not supported and may result in inconsistent results.</p> <p>For 'range':</p> <ul style="list-style-type: none"> <li>• The boundaries themselves are parsed as Double</li> </ul> <p>Example:  <code>'0:10,10:100.89,100: '</code></p> <ul style="list-style-type: none"> <li>• The lower-bound is inclusive. The upper-bound is exclusive. For example, for a facet value corresponding to '0:1000', the constraint should be (attribute <math>\geq 0</math> and attribute <math>&lt;1000</math>).</li> </ul>

Strategy	Description	Options (properties)
alpharange	<p>Applies to string values (since xPlore 1.3). Range grouping requires the use of the 'range' property, which defines the ranges to use for grouping.</p> <p>Example:&lt;\para&gt;</p> <pre> 1 2 3 'a:m,n:r,s:z'</pre> <p>This grouping strategy uses Unicode order, not language dependent order.</p> <p>In practice, it is only useful for ASCII characters.</p>	<p>The range that is mentioned in the range strategy above is also required for the alpharange strategy. In addition to this, the following also apply to the alpharange strategy:</p> <ul style="list-style-type: none"> <li>• The boundaries are parsed as one character. Example: 'a:m,n:r,s:z'</li> <li>• This grouping strategy is case-insensitive. Special care must be taken when adding a constraint based on such a facet value because the comparison operators on string (&lt;, &lt;=, &gt;=, &gt;) are case sensitive. As a consequence, one has to add 2 constraints: one on the lowercase range and one on the uppercase range.</li> </ul> <p>The upper-bound and lower-bound are returned as a property of the facet value. The boundaries are inclusive. As a consequence, the constraint on the upper-bound should add 1 to the actual character.</p> <p>Example: for a facet value corresponding to 'a-m', the constraint should be (attribute &gt;= 'a' and attribute &lt; 'n') or (attribute &gt;='A' and attribute &lt; 'N')</p> <p>For more information, see <a href="#">range</a> on page 663.</p>

Strategy	Description	Options (properties)
The date strategy including: <ul style="list-style-type: none"><li>• day</li><li>• week</li><li>• month</li><li>• quarter (since xPlore v1.3)</li><li>• year</li><li>• relativeDate</li></ul>	The facet values returned are the start date of the range, using the UTC time zone  For example, for January 2011, the facet value is 2001-01-01T00:00:00 (the time is adjusted when the <timezone> property is specified).	<ul style="list-style-type: none"> <li>• <code>timezone</code> (optional): a timezone code. Default: Z (UTC). The timezone to be used to compute date grouping.</li> <li>• <code>returnUTC</code> (optional) - since xPlore 1.3: a boolean. Default: 'true' in REST. When <code>timezone</code> is specified and <code>&lt;returnUTC&gt;</code> is <code>false</code>, the dates in the facets are returned formatted in the specified timezone. The same timezone is used to compute the date range boundaries and to format the facets. When <code>&lt;returnUTC&gt;</code> is set to 'true', the date in the facet is formatted in UTC. In this case, the timezone is only used for computation, but not for formatting.</li> <li>• <code>skipEmptyValues</code> (optional): A boolean value whose default is 'true'. When <code>&lt;skipEmptyValues&gt;</code> is set to 'true', only the facet values with a count strictly greater than zero are returned. When <code>&lt;skipEmptyValues&gt;</code> is set to '<code>false</code>', the query returns all the facet values between the oldest non-zero facet value and the most recent non-zero facet value.</li> </ul> <p> <b>Note:</b> This property only works for date type with grouping strategy day, month, year, week or quarter.</p>

### Max for Facet Group

The `max-values` property is the attribute of the facet definition that defines the greatest number of groups that each facet can return.

### Facet Ordering

- FREQUENCY

Ordered by count, descending

- NONE

Keep the order returned by the facet handler

- VALUE\_ASCENDING

Ordered by facet value ID (ascending). The sort is alphanumeric. We assume that even if the data type of the facet is date or numeric, the values are encoded in such way that they are alphanumerically sortable.

- VALUE\_DESCENDING

Ordered by facet value ID (descending). The sort is alphanumeric. We assume that even if the data type of the facet is date or numeric, the values are encoded in such way that they are alphanumerically sortable.

► **Example 2-274: JSON single facet**

To navigate the facet results, you can POST the original request body to the link in the facet value. The following JSON sample uses a Request and a Response to demonstrate how to use a single facet:

*Request*

```
{  
    ...  
    ...  
    "columns": ["owner_name"],  
    "facet-definitions": [  
        {  
            "id": "id1",  
            "attributes": ["r_object_type"],  
            "sort": "FREQUENCY"  
        }  
    ]  
}
```

*Response*

In the Response, the URL in the facet value has the `facet-id-constraints` parameter. This parameter specifies the facet constraint `id1=dm_folder`:

```
"facets": [  
    {  
        "facet-id": "id1",  
        "facet-label": "Type",  
        "facet-value": [  
            {  
                "facet-id": "id1",  
                "facet-value-constraint": "dm_document",  
                "facet-value-count": 116,  
                "facet-value-id": "dm_document",  
                "link": {  
                    "href": "http://localhost:8080/dctm-rest/repositories/acme01/  
searches/  
0c10304585007313/execution?facet-id-constraints=id1%3Ddm_document",  
                    "rel": "search"  
                }  
            },  
            {  
                "facet-id": "id1",  
                "facet-value-constraint": "dm_folder",  
                "facet-value-count": 68,  
                "facet-value-id": "dm_folder",  
                "link": {  
                    "href": "http://localhost:8080/dctm-rest/repositories/acme01/  
searches/  
0c10304585007313/execution?facet-id-constraints=id1%3Ddm_folder",  
                    "rel": "search"  
                },  
            }  
        ]  
    }  
]
```



### Example 2-275: XML multiple facets

This code sample shows how to use two separate facets against `<r_object_type>` and `<owner_name>`. The facets results have two facet aggregations respectively.

#### XML request

```
<search all-versions="true">
  <facet-definitions>
    <facet-definition id="id1" group-by="alpharange" max-values="8">
      <attributes>
        <attribute>r_object_type</attribute>
      </attributes>
      <properties>
        <property name="range">a:r, s:z</property>
      </properties>
      <sort>FREQUENCY</sort>
    </facet-definition>
    <facet-definition>
      <attributes>
        <attribute>owner_name</attribute>
      </attributes>
    </facet-definition>
  </facet-definitions>
</search>
```

#### JSON response

```
"facets": [
  {
    "facet-id": "id1",
    "facet-label": "Type",
    "facet-value": [
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/Oc10304585007313/execution?facet-id-constraints=id1%3Da:r",
          "rel": "search"
        },
        "facet-id": "id1",
        "facet-value-constraint": "a:r",
        "facet-value-count": 116,
        "facet-value-id": "a:r"
      },
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/Oc10304585007313/execution?&facet-id-constraints=id1%3Ds:z",
          "rel": "search"
        },
        "facet-id": "id1",
        "facet-value-constraint": "s:z",
        "facet-value-count": 68,
        "facet-value-id": "s:z"
      }
    ],
    "facet-owner-name": "Owner Name",
    "facet-label": "Owner Name",
    "facet-value": [
      {
        "link": {
          "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/"
        }
      }
    ]
  }
]
```

```
        "rel": "search"
    },
    "facet-id": "facet_owner_name",
    "facet-value-constraint": "dave",
    "facet-value-count": 126,
    "facet-value-id": "dave"
},
{
    "link": {
        "href": "http://localhost:8080/dctm-rest/repositories/acme01/searches/0c10304585007313/execution?facet-id-constraints=facet_owner_name%3Ddadmin",
        "rel": "search"
    },
    "facet-id": "facet_owner_name",
    "facet-value-constraint": "dmadmin",
    "facet-value-count": 138,
    "facet-value-id": "dmadmin"
}
]
}
```



## Example 2-276: JSON nested facet

This code sample shows a nested facet definition, where `r_object_type` points to `owner_name` in a tree like structure. For each `r_object_type`, the grouping by `owner_name` goes a step further. Also, the facet values are hierarchical. The first nested facet value has constraint `id1=dm_document` and `id2=dmadmin`:

### *JSON request*

```
{  
  "columns": ["owner_name"],  
  "facet-definitions": [  
    {  
      "id": "id1",  
      "attributes": ["r_object_type"],  
      "sort": "FREQUENCY",  
      "facet-definition": {  
        "id": "id2",  
        "attributes": ["owner_name"],  
        "sort": "FREQUENCY"  
      }  
    }  
  }  
}
```



```
{"facets": [
  {
    "facet-id": "id1",
    "facet-label": "Type",
    "facet-value": [
      {
        "facet-value-constraint": "dm_document",
        "facet-value-count": 116,
        "facet-value-id": "dm_document",
        "facet-id": "id1",
        "link": {
          "label": "View"
        }
      }
    ]
  }
]
```

```

        "href": "http://localhost:8080/dctm-rest/repositories/acme01/
searches/
0c10304585007313/execution&facet-id-constraints=id1%3Ddm_document",
        "rel": "search"
    },
    "facet-value": [
        {
            "link": {
                "href": "http://localhost:8080/dctm-rest/repositories/acme01/
searches/0c10304585007313/execution&facet-id-constraints=id1%3D
dm_document,id2%3Ddave",
                "rel": "search"
            },
            "facet-id": "id2",
            "facet-value-constraint": "dave",
            "facet-value-count": 86,
            "facet-value-id": "dave"
        },
        {
            "link": {
                "href": "http://localhost:8080/dctm-rest/repositories/acme01/
searches/0c10304585007313/execution&
facet-id-constraints=id1%3Ddm_document,id2%3Ddmadmin",
                "rel": "search"
            },
            "facet-id": "id2",
            "facet-value-constraint": "dmadmin",
            "facet-value-count": 30,
            "facet-value-id": "dmadmin"
        }
    ]
},
{
    "facet-value-constraint": "dm_folder",
    "facet-value-count": 68,
    "facet-value-id": "dm_folder",
    "facet-id": "id1",
    "link": {
        "href": "http://localhost:8080/dctm-rest/repositories/acme01/
searches/
0c10304585007313/execution&facet-id-constraints=id1%3Ddm_folder",
        "rel": "search"
    },
    "facet-value": [
        {
            "link": {
                "href": "http://localhost:8080/dctm-rest/repositories/acme01/
searches/0c10304585007313/execution&
facet-id-constraints=id1%3Ddm_folder,id2%3Ddave",
                "rel": "search"
            },
            "facet-id": "id2",
            "facet-value-constraint": "dave",
            "facet-value-count": 68,
            "facet-value-id": "dave"
        }
    ]
}
}

```

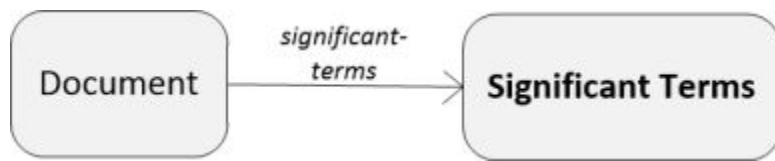


## 2.67 Significant Terms

The *Significant Terms* resource forms part of the services of the Documentum Contextual Content Engine. This resource delivers a way for us to discover common and significant terms by comparing the term frequency change in the requested document and background.

### 2.67.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Document](#)” on page 214 and [About the Diagram](#) on page 13.

### 2.67.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
- Feed Title: Significant Terms of <document name> from CCE
- Supports POST: No
- Feed Updated: Current server time
- Entry: No resource
- Entry ID: The Resource URI
- Entry Title: The significant terms of the document
- Entry Updated: Current server time

### 2.67.3 Link relations

Link relation	Description	Resource reference
self	This collection of significant terms	<a href="#">“Significant Terms” on page 670</a>
search [1]	A template link formed by keyword to search	<a href="#">“Search” on page 645</a>
[1] The above link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/.</code>		

## 2.67.4 Operations

### 2.67.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the significant terms of the requested document.

### 2.67.4.2 Get Significant Terms

Retrieves the significant terms of the requested document.

#### 2.67.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters
GET	page, items-per-page, links

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.67.4.2.2 Request media types

N/A

#### 2.67.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.67.4.2.4 Request body

N/A

#### 2.67.4.2.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.67.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json
- application/hal+json

#### 2.67.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.67.4.2.8 Response body

##### ➡ Example 2-277: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
        090000018000115b/significant-terms</id>
    <title>Significant Terms of document(s) 090000018000115b from CCE</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2018-08-20T03:56:38.419+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        100
    </dm:items-per-page>
    <dm:total xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        2
    </dm:total>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/documents/
              090000018000115b/significant-terms"></link>
    <link rel="http://identifiers.emc.com/linkrel/search"
          hreftemplate="http://localhost:8080/dctm-rest/repositories/REPO/
              search?{collections,facet,inline,
              items-per-page,locations,object-type,page,q,sort,
              timezone,view}"></link>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
            090000018000115b/significant-terms</id>
        <title>desires</title>
        <summary>Significant Term Result</summary>
        <updated>2018-08-20T03:56:38.420+00:00</updated>
        <published>2018-08-20T03:56:38.420+00:00</published>
        <relevance:score xmlns:relevance="http://a9.com/-/opensearch/
            extensions/relevance/1.0/">
            4.058979821368136
        </relevance:score>
        <link rel="http://identifiers.emc.com/linkrel/search"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
                  search?q=desires"></link>
        <content type="application/xml"
                 src="http://localhost:8080/dctm-rest/repositories/REPO/"/>
```

```

        documents/090000018000115b/significant-terms"></content>
    </entry>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/documents/
            090000018000115b/significant-terms</id>
        <title>costing</title>
        <summary>Significant Term Result</summary>
        <updated>2018-08-20T03:56:38.420+00:00</updated>
        <published>2018-08-20T03:56:38.420+00:00</published>
        <relevance:score xmlns:relevance="http://a9.com/-/opensearch/
            extensions/relevance/1.0/">
            4.058979821368136
        </relevance:score>
        <link rel="http://identifiers.emc.com/linkrel/search"
            href="http://localhost:8080/dctm-rest/repositories/REPO/
                search?q=costing"></link>
        <content type="application/xml"
            src="http://localhost:8080/dctm-rest/repositories/REPO/
                documents/090000018000115b/significant-terms"></content>
    </entry>
</feed>
```



### Example 2-278: JSON Response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
        090000018000115b/significant-terms",
    "title": "Significant Terms of document(s) 090000018000115b from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-08-20T03:45:30.694+00:00",
    "page": 1,
    "items-per-page": 100,
    "total": 2,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
                090000018000115b/significant-terms"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/search",
            "hreftemplate": "http://localhost:8080/dctm-rest/repositories/REPO/
                search{?collections,facet,inline,
                items-per-page,locations,object-type,page,q,sort,
                timezone,view}"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
                090000018000115b/significant-terms",
            "title": "desires",
            "summary": "Significant Term Result",
            "updated": "2018-08-20T03:45:30.695+00:00",
            "published": "2018-08-20T03:45:30.695+00:00",
            "score": 4.058979821368136,
            "links": [
                {
                    "rel": "http://identifiers.emc.com/linkrel/search",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                        search?q=desires"
                }
            ]
        }
    ]
}
```

```

        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115b/significant-terms"
        }
    },
    {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
090000018000115b/significant-terms",
        "title": "costing",
        "summary": "Significant Term Result",
        "updated": "2018-08-20T03:45:30.695+00:00",
        "published": "2018-08-20T03:45:30.695+00:00",
        "score": 4.058979821368136,
        "links": [
            {
                "rel": "http://identifiers.emc.com/linkrel/search",
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
search?q=costing"
            }
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115b/significant-terms"
        }
    }
]
}

```



### Example 2-279: HAL: JSON Response

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
090000018000115c/significant-terms",
    "title": "Significant Terms of document(s) 090000018000115c from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-09-14T02:08:10.926+00:00",
    "page": 1,
    "items-per-page": 100,
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/
documents/090000018000115c/significant-terms",
                "title": "profitability",
                "summary": "Significant Term Result",
                "updated": "2018-09-14T02:08:10.927+00:00",
                "published": "2018-09-14T02:08:10.927+00:00",
                "score": 5.88726916052116,
                "links": {
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/documents/090000018000115c/
significant-terms"
                    },
                    "http://identifiers.emc.com/linkrel/search": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/search?q=profitability"
                    }
                }
            },
            ...
        ]
    }
},

```

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/
        documents/090000018000115c/significant-terms",
  "title": "resource",
  "summary": "Significant Term Result",
  "updated": "2018-09-14T02:08:10.927+00:00",
  "published": "2018-09-14T02:08:10.927+00:00",
  "score": 4.03863332553002,
  "_links": {
    "self": {
      "href": "http://localhost:8080/dctm-rest/repositories/
              REPO/documents/090000018000115c/
              significant-terms"
    },
    "http://identifiers.emc.com/linkrel/search": {
      "href": "http://localhost:8080/dctm-rest/repositories/
              REPO/search?q=resource"
    }
  }
},
],
"_links": {
  "self": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              documents/090000018000115c/significant-terms"
  },
  "http://identifiers.emc.com/linkrel/search": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              search{?collections,facet,inline,
              items-per-page,locations,object-type,page,q,sort,
              timezone,view}",
    "templated": true
  }
}
}
```



## 2.68 Sub-Group (s)

### 2.68.1 Sub Group

The *Sub Group* resource represents the relationship between a group and one of its sub groups.

#### 2.68.1.1 Resource relationships

The following diagram illustrates how Sub Group resource is related with other resources:



*See Also:* “Sub Groups” on page 677 and About the Diagram on page 13.

### 2.68.1.2 Feed

No feed support.

### 2.68.1.3 Link relations

N/A

### 2.68.1.4 Operations

#### 2.68.1.4.1 Supported HTTP methods

Method	Description
DELETE	Remove a sub group from a group

#### 2.68.1.4.2 Remove a sub group relationship

Removes the relation between a group and it's sub group.

##### 2.68.1.4.2.1 HTTP Request method and query parameters

DELETE

##### 2.68.1.4.2.2 Request media types

N/A

##### 2.68.1.4.2.3 Request headers

- Accept
- Content-Type
- Authentication

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.68.1.4.2.4 Request body

###### ➡ Example 2-280: Removing a user

```
DELETE http://localhost:8080/dctm-rest/repositories/REPO/groups/  
mygroup/groups/mysubgroup
```



#### 2.68.1.4.2.5 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.68.1.4.2.6 Response media types

N/A

#### 2.68.1.4.2.7 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.68.1.4.2.8 Response body

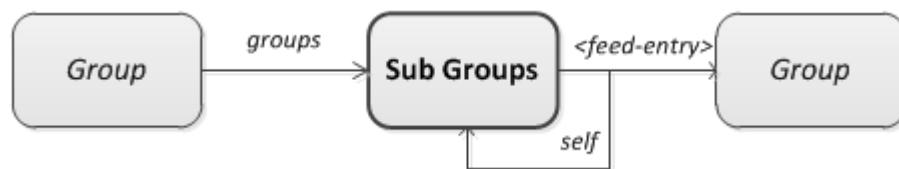
HTTP 204 No Content status upon a successful delete operation.

## 2.68.2 Sub Groups

The *Sub Groups* resource represents a collection of sub groups that directly belong to a specified group.

### 2.68.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Group” on page 308 and [About the Diagram on page 13](#).

### 2.68.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of groups
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Group” on page 308
- Entry ID: URI of the group

- Entry Title: Group name
- Entry Summary: Group description
- Entry Updated: r\_modify\_date of the group

### 2.68.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of groups.	<a href="#">“Groups” on page 317</a>
first, last, next, previous	Pagination links.	<a href="#">“Groups” on page 317</a>

### 2.68.2.4 Operations

#### 2.68.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the information about a collection of groups.

#### 2.68.2.4.2 Add a sub group

Add an existing group into this group as a sub group

##### 2.68.2.4.2.1 HTTP Request method and query parameters

POST

##### 2.68.2.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

##### 2.68.2.4.2.3 Request headers

- Accept
- Content-Type
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.68.2.4.2.4 Request body

Variable	Description	Data type	Value range	Default value
href	The specified group resource URI.  Note: The href variable is mandatory.	string	N/A	null

##### ➡ Example 2-281: XML request

```
<dm:group href="http://localhost:8080/dctm-rest/
repositories/REPO/groups/xxxxx"/>
```



##### ➡ Example 2-282: JSON request

```
{
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/groups/xxxxx"
}
```



#### 2.68.2.4.2.5 Response headers

- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.68.2.4.2.6 Response media types

N/A

#### 2.68.2.4.2.7 Response status

- 201 - Added successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.68.2.4.2.8 Response body**

N/A

#### **2.68.2.4.3 Get Sub Groups**

Retrieve the information about a collection of sub groups for a given parent group.

##### **2.68.2.4.3.1 HTTP Request method and query parameters**

GET

This method supports the following common query parameters:

- inline
- view
- items-per-page
- page
- include-total
- sort
- recursive
- links
- filter

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

##### **2.68.2.4.3.2 Request media types**

N/A

##### **2.68.2.4.3.3 Request headers**

- Accept
- Authorization (optional)

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.68.2.4.3.4 Request body

N/A

#### 2.68.2.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.68.2.4.3.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.68.2.4.3.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.68.2.4.3.8 Response body

XML or JSON representation of the collection of groups.

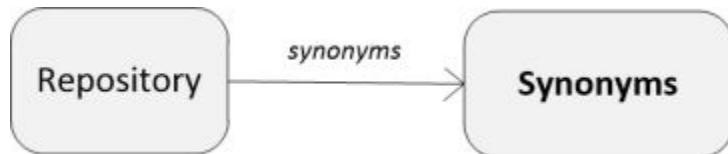
- The body contains a list of the `dm_group` instances (or subtypes of `dm_group`).
- Each object may contain all or a set of properties of the group, depending on the setting of the query parameter `view`.
- The returned child objects collection only contains those that you have access to.
- Pagination is supported.
- By default, the results are listed in alphabetical order by group name.

## 2.69 Synonyms

The *Synonyms* resource is used to get the synonyms of the requested keyword based on the synonym categories in CCE.

### 2.69.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams on page 13](#).

### 2.69.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
- Feed Title: Synonyms from CCE
- Feed Updated: Server's current time
- Supports POST: No
- Entry: No Resource
- Entry ID: The resource URI with a query parameter. For example {repoUri}/cce/synonyms?word=xxx
- Entry Title: The synonyms of the specified word
- Entry Updated: Current server time

### 2.69.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource.	<a href="#">“Synonyms” on page 682</a>
synonym-categories [1]	A link to the Synonym Categories resource.	<a href="#">“Synonym Categories” on page 686</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
<http://identifiers.emc.com/linkrel/>.

## 2.69.4 Operations

### 2.69.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the synonyms from the static dictionary of build-in and customized categories.

### 2.69.4.2 Get Synonyms

This operation gets the synonyms of the requested words.

#### 2.69.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	word	The word whose synonyms you want to retrieve.	String	Any string	null
	category	The Synonym Category that you want to search within.	String	A category from the Synonym Categories resource	null

#### 2.69.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.69.4.2.3 Request media types**

N/A

**2.69.4.2.4 Request body**

N/A

**2.69.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.69.4.2.6 Response headers**

Content-Type

Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.69.4.2.7 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

**2.69.4.2.8 Response body** **Example 2-283: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/cce/synonyms</id>
    <title>Synonyms from CCE</title>
    <author>
      <name>OpenText Documentum</name>
    </author>
    <updated>2018-09-25T02:09:42.528+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
      100
    </dm:items-per-page>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/cce/
                synonyms?word=openText"></link>
    <link rel="http://identifiers.emc.com/linkrel/synonym-categories"
          href="http://localhost:8080/dctm-rest/repositories/REPO/cce/"/>
```

```

        synonym-categories"></link>
<entry>
<id>http://localhost:8080/dctm-rest/repositories/REPO/cce/
    synonyms?word=opentext</id>
<title>qi xin</title>
<summary>
    Synonym Result from Category Enterprise_Content_Management
</summary>
<updated>2018-09-25T02:09:42.529+00:00</updated>
<published>2018-09-25T02:09:42.529+00:00</published>
<content type="application/vnd.emc.documentum+xml"
    src="http://localhost:8080/dctm-rest/repositories/REPO/cce/
        synonyms?word=opentext"></content>
</entry>
</feed>

```



#### Example 2-284: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/synonyms",
    "title": "Synonyms from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-08-20T04:53:28.973+00:00",
    "page": 1,
    "items-per-page": 100,
    "total": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                synonyms?word=opentext"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/synonym-categories",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                synonym-categories"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                synonyms?word=opentext",
            "title": "Enterprise_Content_Management",
            "summary": "Significant Term Result from Category qi xin",
            "updated": "2018-08-20T04:53:28.974+00:00",
            "published": "2018-08-20T04:53:28.974+00:00",
            "content": {
                "type": "application/vnd.emc.documentum+json",
                "src": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                    synonyms?word=opentext"
            }
        }
    ]
}
```



 **Example 2-285: HAL: JSON Response**

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/synonyms",
    "title": "Synonyms from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-09-17T06:19:14.236+00:00",
    "page": 1,
    "items-per-page": 100,
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonyms?word=opentext",
                "title": "qi xin",
                "summary": "Synonym Result from
Category Enterprise_Content_Management",
                "updated": "2018-09-17T06:19:14.236+00:00",
                "published": "2018-09-17T06:19:14.236+00:00",
                "_links": {
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/cce/synonyms?word=opentext"
                    }
                }
            }
        ]
    },
    "_links": {
        "http://identifiers.emc.com/linkrel/synonym-categories": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonym-categories"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonyms?word=opentext"
        }
    }
}
```

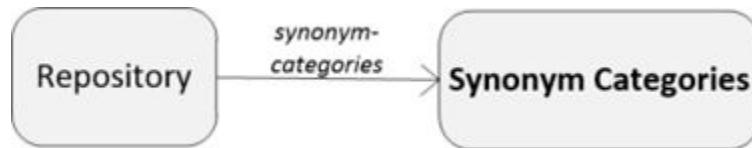


## 2.70 Synonym Categories

The *Synonym Categories* resource is used to get the static synonym categories from the built-in dictionaries of the Intellisophic taxonomy definitions and customized ones in CCE.

## 2.70.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams](#) on page 13.

## 2.70.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
- Feed Title: Synonym Categories from CCE
- Feed Updated: Server's current time
- Supports POST: No
- Entry: No Resource
- Entry ID: Resource URI
- Entry Title: The name of the category
- Entry Updated: Current server time

## 2.70.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource.	“ <a href="#">Synonym Categories</a> ” on page 686
synonyms [1]	A link to the Synonyms resource.	“ <a href="#">Synonyms</a> ” on page 682

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <http://identifiers.emc.com/linkrel/>.

## 2.70.4 Operations

### 2.70.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the static synonym categories of the build-in and customized categories.

### 2.70.4.2 Get Synonym Categories

This operation retrieves the Synonym categories.

#### 2.70.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.70.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.70.4.2.3 Request media types

N/A

#### 2.70.4.2.4 Request body

N/A

#### 2.70.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.70.4.2.6 Response headers

Content-Type

Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.70.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.70.4.2.8 Response body

##### Example 2-286: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>
    http://localhost:8080/dctm-rest/repositories/REPO/cce/synonym-categories
  </id>
  <title>Synonym Categories from CCE</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2018-08-20T04:34:56.706+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
  </dm:items-per-page>
  <dm:total xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    21
  </dm:total>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/cce/
              synonym-categories"></link>
  <link rel="http://identifiers.emc.com/linkrel/synonyms"
        href="http://localhost:8080/dctm-rest/repositories/REPO/cce/
              synonyms"></link>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/cce/
          synonym-categories
    </id>
    <title>cdisc_send_terminology</title>
    <summary>Result from CCE</summary>
    <updated>2018-08-20T04:34:56.706+00:00</updated>
    <published>2018-08-20T04:34:56.706+00:00</published>
    <link rel="search"
          href="http://localhost:8080/dctm-rest/repositories/REPO/
              search?q=cdisc_send_terminology"></link>
    <content type="application/xml"
             src="http://localhost:8080/dctm-rest/repositories/REPO/cce/
                  synonym-categories"></content>
  </entry>
  <entry>
```

```
<id>http://localhost:8080/dctm-rest/repositories/REPO/cce/
    synonym-categories
</id>
<title>legal_terminology</title>
<summary>Result from CCE</summary>
<updated>2018-08-20T04:34:56.707+00:00</updated>
<published>2018-08-20T04:34:56.707+00:00</published>
<link rel="search"
      href="http://localhost:8080/dctm-rest/repositories/REPO/
          search?q=legal_terminology"></link>
<content type="application/xml"
          src="http://localhost:8080/dctm-rest/repositories/REPO/cce/
              synonym-categories"></content>
</entry>
</feed>
```



### Example 2-287: JSON response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
    synonym-categories",
  "title": "Synonym Categories from CCE",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2018-08-20T04:34:07.805+00:00",
  "page": 1,
  "items-per-page": 100,
  "total": 2,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
        synonym-categories"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/synonyms",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
        synonyms"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
        synonym-categories",
      "title": "cdisc_send_terminology",
      "summary": "Result from CCE",
      "updated": "2018-08-20T04:34:07.806+00:00",
      "published": "2018-08-20T04:34:07.806+00:00",
      "links": [
        {
          "rel": "search",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            search?q=cdisc_send_terminology"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
          cce/synonym-categories"
      }
    },
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
```

```

        "synonym-categories",
        "title": "legal_terminology",
        "summary": "Result from CCE",
        "updated": "2018-08-20T04:34:07.806+00:00",
        "published": "2018-08-20T04:34:07.806+00:00",
        "links": [
            {
                "rel": "search",
                "href": "http://localhost:8080/dctm-rest/repositories/
REPO/search?q=legal_terminology"
            }
        ],
        "content": {
            "type": "application/vnd.emc.documentum+json",
            "src": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonym-categories"
        }
    }
}

```



#### ➡ Example 2-288: HAL: JSON Response

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonym-categories",
    "title": "Synonym Categories from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-09-25T02:04:40.172+00:00",
    "page": 1,
    "items-per-page": 100,
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonym-categories",
                "title": "cdisc_send_terminology",
                "summary": "Result from CCE",
                "updated": "2018-09-25T02:04:40.173+00:00",
                "published": "2018-09-25T02:04:40.173+00:00",
                "links": {
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/cce/synonym-categories"
                    },
                    "http://identifiers.emc.com/linkrel/search": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/search?q=cdisc_send_terminology"
                    }
                }
            },
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonym-categories",
                "title": "legal_terminology",
                "summary": "Result from CCE",
                "updated": "2018-09-25T02:04:40.173+00:00",
                "published": "2018-09-25T02:04:40.173+00:00",
                "links": {
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
REPO/cce/synonym-categories"
                    }
                }
            }
        ]
    }
}

```

```

        },
        "http://identifiers.emc.com/linkrel/search": {
            "href": "http://localhost:8080/dctm-rest/repositories/
REPO/search?q=legal_terminology"
        }
    }
}
],
"_links": {
    "http://identifiers.emc.com/linkrel/synonyms": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonyms"
    },
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
synonym-categories"
    }
}
}

```

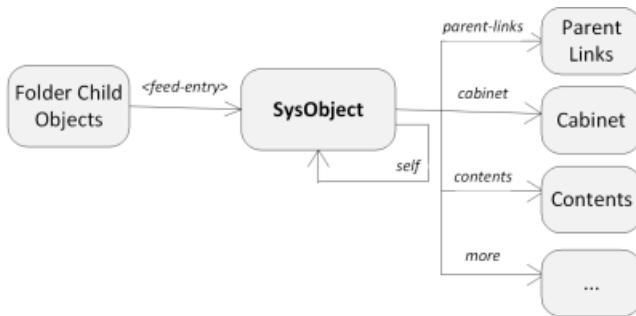


## 2.71 SysObject

The SysObject resource represents a SysObject instance in a repository, such as a Document, or a Folder.

### 2.71.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Folder Child Objects](#)” on page 282 and [About the Diagram](#) on page 13.

## 2.71.2 Feed

No feed support.

## 2.71.3 Link relations

*Any SysObject*

Link relation	Description	Resource reference
self	This SysObject resource	<a href="#">"SysObject" on page 692</a>
parent-links [1]	Parent folder (or cabinet) of this SysObject resource	<a href="#">"Parent Links" on page 499</a>
edit [1]	Edit this object	<a href="#">"SysObject" on page 692</a>
delete [1]	Delete this object	<a href="#">"SysObject" on page 692</a>
relations [1]	Collection of relations that are related to this SysObject resource	<a href="#">"Relations" on page 538</a>
comments [1]	The comments associated to this object	<a href="#">"Comments" on page 156</a>
virtual-document-nodes [1]	The virtual document nodes of a Virtual Document SysObject	<a href="#">"Virtual Document Nodes" on page 821</a>
virtual-document-conversion	Convert a simple document into a Virtual Document by adding a new node to it. This link is only displayed when the object is not already a virtual document.	<a href="#">"Virtual Document Nodes" on page 821</a>
object-aspects [1]	Attached aspect of the SysObject	<a href="#">"SysObject" on page 692</a>
permission-set [1]	Permission set of the SysObject	<a href="#">"Permission Set" on page 507</a>
permissions [1]	Specified user's permissions on a given SysObject	<a href="#">"Permissions" on page 503</a>
snapshots [1]	The snapshots associated with this virtual document. This link is only displayed when the object is a virtual document.	<a href="#">"Virtual Document Snapshots" on page 839</a>
attachable-lifecycles [1]	Collection of Lifecycles to which the given SysObject can be attached	<a href="#">"Lifecycles" on page 385</a>
object-lifecycle [1]	The current Lifecycle state against the given SysObject	<a href="#">"Object Lifecycle" on page 460</a>
[1] The above link relations are defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

*Folder Specific*

<b>Link relation</b>	<b>Description</b>	<b>Resource reference</b>
folders [1]	Collection of child folders that are linked to this <code>SysObject</code> resource	<a href="#">"Folder Child Folders" on page 268</a>
documents [1]	Collection of documents that are linked to this <code>SysObject</code> resource	<a href="#">"Folder Child Documents" on page 240</a>
objects [1]	Collection of <code>SysObject</code> resources that are linked to this <code>SysObject</code> resource	<a href="#">"Folder Child Objects" on page 282</a>
child-links [1]	Collection of folder children	<a href="#">"Folder Child Objects" on page 282</a>
[1] The above link relations are defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/.</code>		

*Any Non-Folder `SysObject`*

<b>Link relation</b>	<b>Description</b>	<b>Resource reference</b>
contents	Collection of relations that is related to this <code>SysObject</code> resource	<a href="#">"Contents" on page 178</a>
primary-content [1]	Points to the primary content metadata of this object	<a href="#">"Content" on page 171</a>
[1] The above link relations are defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/.</code>		

*Versioning Link for Any Non-Folder `SysObject`*

<b>Link relation</b>	<b>Description</b>	<b>Resource reference</b>
checkout [1]	Check out this object	<a href="#">"Lock" on page 416</a>
checkin-next-major [1]	Check in this object with the next major version policy	<a href="#">"All Versions" on page 16</a>
checkin-next-minor [1]	Check in this object with the next minor version policy	<a href="#">"All Versions" on page 16</a>
checkin-branch [1]	Check in this object with the branch version policy	<a href="#">"All Versions" on page 16</a>
cancel-checkout [1]	Cancel the check-out operation on this object.	<a href="#">"Lock" on page 416</a>
version-history	Collection of versions of this document	<a href="#">"All Versions" on page 16</a>
current-version [1]	Current version of the object	<a href="#">"Current Version" on page 205</a>
predecessor-version	Predecessor version of the object	<a href="#">"SysObject" on page 692</a>

Link relation	Description	Resource reference
[1] The above link relations are defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

*Lightweight Object Specific*

Link relation	Description	Resource reference
shared-parent [1]	When the object is a lightweight child object, this is the shared parent of the SysObject	<a href="#">"Lightweight Object Parent" on page 408</a>
lightweight-objects [1]	Primary content metadata	<a href="#">"Lightweight Objects" on page 400</a>
materialize [1]	Materialize the lightweight object following the link relation	<a href="#">"Lightweight Object Materialization" on page 413</a>
dematerialize [1]	Dematerialize the lightweight object following the link relation	<a href="#">"Lightweight Object Materialization" on page 413</a>
[1] The above link relations are defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		



### Caution

- Versioning links for the SysObject collection are not displayed.
- The `checkin-next-major`, `checkin-next-minor`, and `checkin-branch` links are only displayed when their related version policy is allowed.
- The Foundation Java API or Foundation Java API ACL object retrieves the permission dependant [ACL](#) for all of the `<checkin-*>` links, such as `checkin-next-major`, `checkin-next-minor`, and `checkin-branch`, which depend on the Foundation Java API object and version policy. When using any of the above links, the system retrieves the objects one by one to allow Foundation Java API to verify permissions.
- You must have Version user privileges to use the `<cancel-checkout>` link.
- The `<browse>` (allows you to view metadata only), `<read>` (allows you to view content and metadata), `<delete>`, and `<write>` permissions depend on the [ACL](#) and can be retrieved using either the Foundation Java API object, or the Foundation Java API ACL object.

*Permission-Enabled*

Link relation	Basic permission	Extended permission	Object state	User privilege	Version policy
self	browse				

Link relation	Basic permission	Extended permission	Object state	User privilege	Version policy
contents	read				
primary-content	read				
edit	write		not locked	lock owner if the object is locked	
delete	delete	delete	not locked		
child-links	browse				
parent-links	browse				
checkout	version		not locked		
checkin-next-major	version				allowed next major
checkin-next-minor	version		checked out	lock owner	allowed next minor
checkin-branch	version			lock owner	allowed branch
cancel-checkout	version			lock owner, object owner, or super user	
version-history	browse				
current-version	browse				
predecessor-version	browse				

When working with SysObjects the following items apply:

- *edit*: When the object is locked, then only the lock owner (with write permissions) can edit the object.
- *delete*: This can only be performed on a locked object and must allow versioning.
- *checkout*: The object is not locked and has versioning permissions.
- *checkin*: This can only be performed when the lock owner has versioning permissions and the object version allows a version policy.
- *cancel-checkout*: This can only be performed by a user who has versioning permissions, is the lock owner, is the object owner, or by the super user.

#### *Difference Between a Single SysObject and a SysObject Collection Feed*

A single `SysObject` resource differs from a collection feed of the `SysObject` resource in the following aspects:

- A single `SysObject` contains versioning links while a collection does not.
- Foundation REST API does not perform any permission checks on a collection
- The contents links are shown in a collection without a permission check.
- When an object is not locked in a collection, the `edit` and `delete` links are displayed without a permission check, meaning that even when a user is not the lock owner or does not have the corresponding permissions, the `edit` and `delete` links are accessible to that user.
- When an object is locked, there's no `delete` link in either the single or collection resource.
- When an object is locked in a collection, and a user is the lock owner, the `edit` link is displayed without a permission check, meaning that even when the user does not have the `write` permission, the `edit` link is accessible to that user.
- When a single resource is locked and the user is the lock owner. In this case, a check to determine if the user has `write` permissions is done.
- When a collection resource is locked and the user is the lock owner, the `edit` link is always displayed, without any permission checking.



**Tip:** A simple performance test to retrieve a 100 items per page has determined that retrieving the Foundation Java API object, calculating the `<ACL>` and version policy, is up to 6 times slower than retrieving the collection using DQL with attributes. This latency may be worsened when using a WAN because of the numerous RPC calls that are necessary.

To help with the above latency, avoid checking permissions when working with a collection. When enabling a link requires the use of other objects in addition to the current object, then the permissions are not checked. For example, when working with a parent and cabinet link relation, the parent folder and cabinet do not perform any checks to verify access.

## 2.71.4 Operations

### 2.71.4.1 Supported HTTP methods

Method	Description
GET	Retrieves properties, and other information of the <code>SysObject</code> resource
POST	Updates the attributes of the <code>SysObject</code> resource
DELETE	Deletes the <code>SysObject</code> instance from a repository

### 2.71.4.2 Get an object

GET attributes, permissions, and other information on the object instance. Attributes and permissions are returned as embedded elements in the Response message body. Other information, such as relationships, versions, and contents, are referenced using link relations in the Response message body.

#### 2.71.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	view				
	check-subscription	Determines whether or not to check if the current user is subscribed to the requested instance.	string	<i>true, false</i>	<i>false</i>

#### 2.71.4.2.2 Request media types

N/A

#### 2.71.4.2.3 Request headers

- Accept
- Authorization (optional)

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.71.4.2.4 Request body

N/A

#### 2.71.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.71.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.71.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.71.4.2.8 Response body

##### ➡ Example 2-289: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<object xmlns="http://identifiers.emc.com/vocab/documentum"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="dm_sysobject"
         definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_sysobject">
    <properties>
        <attr1>value1</attr1>
        <repeating_attr2>
            <item href="xxx2">value2</item>
            <item href="xxx3">value3</item>
        </repeating_attr2>
        ...
    </properties>
    <link rel="self" href="..."/>
    ...
</object>
```



##### ➡ Example 2-290: JSON response

```
{
    "name": "object",
    "type": "dm_document",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject",
    "properties": {
        {
            "attr1": "value1",
            "repeating_attr2": [value2, value3],
            ...
        },
        "links": [
            {
                "rel": "self",
                "href": "..."
            },
            ...
        ]
    }
}
```



#### 2.71.4.2.9 Example



##### Example 2-291: JSON request - Get an object

```
GET /dctm-rest/repositories/REPO/objects/0900000580002516?view=object_name,
      i_folder_id HTTP/1.1
Host: localhost:8080
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
Accept: application/vnd.emc.documentum+json
```



##### Example 2-292: JSON response - GET an object

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "name": "object",
  "type": "dm_document",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document",
  "properties": {
    "object_name": "iig36_Wed_Gal1005_1000_MurphyOil.pptx",
    "i_folder_id": [
      "0c0000058000211b"
    ]
  },
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516?view=object_name,i_folder_id"
    },
    {
      "rel": "edit",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/delete",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516"
    },
    {
      "rel": "canonical",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/documents/
0900000580002516"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/parent-links",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/parent-links"
    },
    {
      "rel": "parent",
      "title": "0c0000058000211b",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/folders/
0c0000058000211b"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/cabinet",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/cabinets/
```

```

        "0c0000058000211b"
    },
    {
        "rel": "contents",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/contents"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/primary-content",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/contents/content"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/checkout",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/lock"
    },
    {
        "rel": "version-history",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/versions"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/current-version",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/versions/current"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/virtual-document-nodes",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/vd-nodes"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/snapshots",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots?
filter=r_assembled_from_id%20%3D'0900000580002516'"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/relations",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/relations?related-
object-id=0900000580002516&related-object-role=any"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/object-aspects",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/aspects"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/permission-set",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/permission-set"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/permissions",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/permissions"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/comments",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580002516/comments"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/attachable-lifecycles",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/lifecycles?object-
id=0900000580002516"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/object-lifecycle",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/

```

```
    0900000580002516/lifecycle"
    }
}
```



### 2.71.4.3 Update an object

You can use this operation to update a SysObject resource by providing new values for attributes.



#### Caution

Providing undefined attributes or invalid values for attributes in the Request body causes an exception to be thrown.

#### 2.71.4.3.1 HTTP Request method and query parameters

POST

#### 2.71.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.71.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.71.4.3.4 Request body

Only attributes that can be updated should be included in the request message body. You cannot use this operation to make updates to permissions.



#### Example 2-293: XML request

```
<?xml version="1.0" encoding="utf-8"?>
<object xmlns="http://identifiers.emc.com/vocab/documentum"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <properties>
        <attr1>value1</attr1>
        <repeating_attr2>
            <item href="xxx2">value2</item>
            <item href="xxx3">value3</item>
        </repeating_attr2>...</properties>
    </object>
```



▶ **Example 2-294: JSON request**

```
{
  "properties":
  {
    "attr1": "value1",
    "repeating_attr2": [value2, value3],
    ...
  }
}
```



#### 2.71.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.71.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.71.4.3.7 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.71.4.3.8 Response body

XML or JSON representation of the object. A successful response contains all of the attributes of the updated object. Permission set is not returned.

▶ **Example 2-295: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<object xmlns="http://identifiers.emc.com/vocab/documentum"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="dm_sysobject"
         definition="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject">
  <dm:attributes>
    <dm:attr1>value1</dm:attr1>
    <dm:repeating_attr2>
      <dm:item href="xxx2">value2</dm:item>
      <dm:item href="xxx3">value3</dm:item>
    </dm:repeating_attr2>
    ...
  </dm:attributes>
```

```
<link rel="self" href="..." />
...
</dm:dm_document>
```



#### Example 2-296: JSON response

```
{
  "properties":
  {
    "attr1": "value1",
    "repeating_attr2": [value2, value3],
    ...
  },
  "links":
  [
    {
      "rel": "self",
      "href": ...
    },
    ...
  ]
}
```



#### 2.71.4.3.9 Example

##### Example 2-297: JSON request

```
POST http://localhost:8080/dctm-rest/repositories/REPO/objects/1900000f80000cd8
Accept:application/vnd.emc.documentum+json
Content-Type:application/vnd.emc.documentum+xml
Authorization:Basic ZG1hZG1pbjpwYXNzd29yZA==

<?xml version="1.0" encoding="UTF-8"?>
<object xmlns="http://identifiers.emc.com/vocab/documentum"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <properties>
    <object_name>dm_usage_log_modified</object_name>
  </properties>
</object>
```



##### Example 2-298: JSON response

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "properties":
  {
    "r_object_id": "1900000f80000cd8",
    "object_name": "dm_usage_log_modified",
    ...
  },
  "links":
  [
    {
      "rel": "self",
    }
  ]
}
```

```

        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
1900000f80000cd8"
},
{
  "rel": "contents",
  "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
1900000f80000cd8/contents"
}
]
}

```



#### 2.71.4.4 Delete an object

Delete the `SysObject` resource from the repository.

For a folder object, the deletion of folder children is controlled by query parameters. When a deletion fails, an exception is thrown and the `SysObject` resource (including the folder tree and version history) is reverted back to its original state.

For a document object, the version policy is controlled by query parameters.

##### 2.71.4.4.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE	del-non-empty	<p>Specifies whether or not this operation deletes a non-empty folder.</p> <ul style="list-style-type: none"> <li><i>true</i> - The folder and all its descendants are deleted.</li> <li><i>false</i> - Only this folder is deleted. Deleting a non-empty folder results in an error.</li> </ul>	string	<i>true, false</i>	<i>false</i>
	del-all-links	<p>Specifies whether a multi-linked descendant is deleted or unlinked from this specified folder.</p> <ul style="list-style-type: none"> <li><i>true</i> - This operation deletes a multi-linked descendant along with all its folder links.</li> <li><i>false</i> - This operation only deletes a descendant's link to this folder. The descendant can still remain linked to other folders.</li> </ul> <p> <b>Note:</b> Virtual documents are not supported.</p>			

HTTP method	Query parameters	Description	Data type	Value range	Default value
	del-version	<p>Deletion options for multi-version document objects.</p> <ul style="list-style-type: none"> <li>selected - The version associated with the given object ID is removed.</li> <li>unused - (Default) The versions that do not have symbolic labels associated are removed.</li> <li>all - All revisions on the same version tree are removed.</li> </ul>	string	N/A	<all>
	del-vd-all	<p>Specifies whether this operation should delete a virtual document and all of its descendants.</p> <p><i>false</i>: Deletes only the virtual document, the containments, and its direct children. It does not delete any other descendants of the virtual document.</p> <p><i>true</i>: Deletes the virtual document and all of the descendants of that virtual document.</p> <p>For more information, see <a href="#">Note</a>.</p>	string	<i>true, false</i>	<i>false</i>

#### 2.71.4.4.2 Request media types

N/A

#### 2.71.4.4.3 Delete a virtual document

When the `del-vd-all` parameter is *true*, it traverses the virtual document and applies the strategy specified by the `del-version` parameter to each node in the hierarchy. The virtual document hierarchy depends on the following:

- The criteria defined for each node when it was added or last updated.
- The value of the version data that is present in the `resolution_label` attribute of the virtual document object being deleted. When the `resolution_label` attribute exists; the virtual document uses it to determine the version. Otherwise, the virtual document must use `CURRENT` to retrieve the virtual document hierarchy. When the virtual document uses `CURRENT` to retrieve the virtual document hierarchy, late binding nodes may have their version data, which is specified by the `resolution_label`, deleted.
- The `include-broken` attribute is always set to *true* when determining the virtual document hierarchy.

The following table displays the strategies that can be used to delete nodes.

<b>del-vd-all Value</b>	<b>del-version Strategy</b>	<b>Transactional</b>	<b>Non-transactional</b>
<i>true</i>	all	If an object appears in the Virtual Document hierarchy more than once, it does not get the object more than one time and the transaction will fail.	Delete what can be deleted and leave the other existing nodes.
	selected	If the virtual document object has multiple versions, its nodes have virtual document relationship with all of these objects.  Deletion will fail because the nodes are referenced by the unselected virtual document object.	
	unused	Always fail for simplest virtual document structure.	

#### 2.71.4.4.4 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.71.4.4.5 Request body

N/A

#### 2.71.4.4.6 Response headers

N/A

#### 2.71.4.4.7 Response media types

N/A

#### 2.71.4.4.8 Response status

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.71.4.4.9 Response body

The following status code indicates a successful deletion:

Status Code: 204 No Content Success

The Response body contains no content.

### 2.71.5 Example

#### ➤ Example 2-299: JSON request

```
DELETE http://localhost:8080/dctm-rest/repositories/REPO/objects/1900000f80000cd8
Accept:application/vnd.emc.documentum+json
Authorization:Basic ZG1hZG1pbjpwYXNzd29yZA==
```



#### ➤ Example 2-300: JSON response

```
Status Code: 204 No Content Success
```



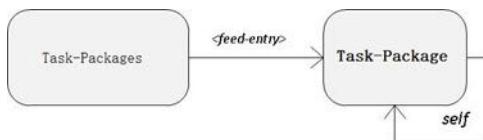
## 2.72 Task Package (s)

### 2.72.1 Task Package

The *Task Package* resource is used to get the package, known as an attachment, of a specified workflow task or to add contents to the package of a specified workflow task.

#### 2.72.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Task Packages” on page 719 and [About the Diagram on page 13](#).

### 2.72.1.2 Feed

No feed support.

### 2.72.1.3 Link relations

Link relation	Description	Resource reference
self	A link to itself	<a href="#">"Task Package" on page 708</a>
delete-documents [1]	To delete all of the documents in a package	<a href="#">"Task Package Documents" on page 723</a>
delete-note [1]	To delete a specified note in a package	<a href="#">"Task Package Note" on page 725</a>
document [1]	To fetch an attached document	<a href="#">"SysObject" on page 692</a>
work-item [1]	To fetch the associated work item	<a href="#">"Work Item" on page 869</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.72.1.4 Operations

#### 2.72.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the task package
POST	Add documents and notes to a task package

#### 2.72.1.4.2 Get a Task Package

Retrieves a task package.

##### 2.72.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### **2.72.1.4.2.2 Request media types**

N/A

#### **2.72.1.4.2.3 Request headers**

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.72.1.4.2.4 Request body**

N/A

#### **2.72.1.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.72.1.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### **2.72.1.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.72.1.4.2.8 Response body

➡ Example 2-301: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<package xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <package-name>DocumentToReview</package-name>
    <status>FULL</status>
    <optional>false</optional>
    <visible>false</visible>
    <operation></operation>
    <required-type>dm_document</required-type>
    <note-count>1</note-count>
    <notes>
        <note>
            <id>410000018005410c</id>
            <text>this is the first note. true-persistent
            </text>
            <creation-date>2019-08-21T06:37:54.000+00:00
            </creation-date>
            <flag>1</flag>
            <persistent>true</persistent>
            <writer>dave</writer>
            <is-new>false</is-new>
        </note>
    </notes>
    <document-count>1</document-count>
    <document-ids>
        <document-id>0900000180011fad</document-id>
    </document-ids>
    <sskill-level></skill-level>
    <form-template-defined>true</form-template-defined>
    <form-for-properties>false</form-for-properties>
    <task-id>4a00000180002900</task-id>
    <index>0</index>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900/packages/0"/>
        <link rel="http://identifiers.emc.com/linkrel/
                    delete-documents"
              href="http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900/packages/0/
                    documents"/>
        <link rel="http://identifiers.emc.com/linkrel/
                    document"
              title="0900000180011fad"
              href="http://localhost:8080/dctm-rest/
                    repositories/REPO/objects/
                    0900000180011fad"/>
        <link rel="http://identifiers.emc.com/linkrel/
                    delete-note"
              title="410000018005410c"
              href="http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900/packages/0/notes/
                    410000018005410c"/>
        <link rel="http://identifiers.emc.com/linkrel/
                    work-item"
              href="http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900"/>
    </links>
  </package>

```



 **Example 2-302: JSON response**

```
{
  "package-name": "DocumentToReview",
  "status": "FULL",
  "optional": false,
  "visible": false,
  "operation": "",
  "required-type": "dm_document",
  "note-count": 1,
  "notes": [
    {
      "id": "410000018005410c",
      "text": "this is the first note. true-persistent",
      "creation-date": "2019-08-21T06:37:54.000+00:00",
      "flag": 1,
      "persistent": true,
      "writer": "dave",
      "is-new": false
    }
  ],
  "document-count": 1,
  "document-ids": [
    "0900000180011fad"
  ],
  "skill-level": 0,
  "form-template-defined": true,
  "form-for-properties": false,
  "task-id": "4a00000180002900",
  "index": 0,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/work-items/
        4a00000180002900/packages/0"
    }, {
      "rel": "http://identifiers.emc.com/linkrel/
        delete-documents",
      "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/work-items/
        4a00000180002900/packages/0/
        documents"
    }, {
      "rel": "http://identifiers.emc.com/linkrel/
        document",
      "title": "0900000180011fad",
      "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/objects/
        0900000180011fad"
    }, {
      "rel": "http://identifiers.emc.com/linkrel/
        delete-note",
      "title": "410000018005410c",
      "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/work-items/
        4a00000180002900/packages/0/notes/
        410000018005410c"
    }, {
      "rel": "http://identifiers.emc.com/linkrel/
        work-item",
      "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/work-items/
        4a00000180002900"
    }
  ]
}
```



▶ Example 2-303: HAL + JSON response

```
{
    "package-name": "DocumentToReview",
    "status": "FULL",
    "optional": false,
    "visible": false,
    "operation": "",
    "required-type": "dm_document",
    "note-count": 1,
    "notes": [
        {
            "id": "410000018005410c",
            "text": "this is the first note. true-persistent",
            "creation-date": "2019-08-21T06:37:54.000+00:00",
            "flag": 1,
            "persistent": true,
            "writer": "dave",
            "is-new": false
        }
    ],
    "document-count": 1,
    "document-ids": [
        "0900000180011fad"
    ],
    "skill-level": 0,
    "form-template-defined": true,
    "form-for-properties": false,
    "task-id": "4a00000180002900",
    "index": 0,
    "_links": {
        "http://identifiers.emc.com/linkrel/document": {
            "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/objects/
                    0900000180011fad",
            "title": "0900000180011fad"
        },
        "http://identifiers.emc.com/linkrel/delete-note": {
            "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900/packages/0/notes/
                    410000018005410c",
            "title": "410000018005410c"
        },
        "http://identifiers.emc.com/linkrel/work-item": {
            "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900/packages/0"
        },
        "http://identifiers.emc.com/linkrel/
        delete-documents": {
            "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/work-items/
                    4a00000180002900/packages/0/
                    documents"
        }
    }
}
```



#### 2.72.1.4.3 Add documents and notes

Adds documents or notes to a package.

##### 2.72.1.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	N/A	N/A			

##### 2.72.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/hal+json

##### 2.72.1.4.3.3 Request headers

- Authorization
- Accept
- Content-Type

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.72.1.4.3.4 Request body

###### ➡ Example 2-304: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<package>
  <notes>
    <note>
      <text>note abc</text>
      <persistent>false</persistent>
    </note>
    <note>
      <text>note def</text>
      <persistent>false</persistent>
    </note>
  </notes>
  <document-ids>
    <document-id>0900000180002379</document-id>
  </document-ids>
</package>
```



###### ➡ Example 2-305: JSON request

```
{
  "document-ids": [
    "090000018000220"
  ],
  "notes": [ {
```

```

        "text": "note abc",
        "persistent": false
    }, {
        "text": "note def",
        "persistent": false
    }
]
}

```



#### ➤ Example 2-306: HAL + JSON request

```
{
    "document-ids": [
        "0900000180000220"
    ],
    "notes": [
        {
            "text": "note abc",
            "persistent": false
        }, {
            "text": "note def",
            "persistent": false
        }
    ]
}
```



**Note:** The following are the only attributes that apply to *notes*, all others are ignored.

- *text*: The note message that is being added
- *persistent*:
  - When *true*, then the note is visible to all subsequent workflow tasks.
  - When *false*, then the note is visible in the next task only

##### 2.72.1.4.3.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.72.1.4.3.6 Response media types

N/A

#### 2.72.1.4.3.7 Response status

- 200 - Successfully added
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.72.1.4.3.8 Response body



**Note:** The Response body for this operation will be the same as the Response body for the *Get a Task Package* operation. It is shown below for your convenience.



##### Example 2-307: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<package xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <package-name>DocumentToReview</package-name>
    <status>FULL</status>
    <optional>false</optional>
    <visible>false</visible>
    <operation></operation>
    <required-type>dm_document</required-type>
    <note-count>1</note-count>
    <notes>
        <note>
            <id>410000018005410c</id>
            <text>this is the first note. true-persistent
            </text>
            <creation-date>2019-08-21T06:37:54.000+00:00
            </creation-date>
            <flag>1</flag>
            <persistent>true</persistent>
            <writer>dave</writer>
            <is-new>false</is-new>
        </note>
    </notes>
    <document-count>1</document-count>
    <document-ids>
        <document-id>0900000180011fad</document-id>
    </document-ids>
    <skill-level>0</skill-level>
    <form-template-defined>true</form-template-defined>
    <form-for-properties>false</form-for-properties>
    <task-id>4a00000180002900</task-id>
    <index>0</index>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/
REPO/work-items/4a00000180002900/packages/
0"/>
        <link rel="http://identifiers.emc.com/linkrel/
delete-documents"
              href="http://localhost:8080/dctm-rest/repositories/
REPO/work-items/4a00000180002900/
packages/0/documents"/>
        <link rel="http://identifiers.emc.com/linkrel/document"
```

```

        title="0900000180011fad"
        href="http://localhost:8080/dctm-rest/repositories/
            REPO/objects/0900000180011fad"/>
    <link rel="http://identifiers.emc.com/linkrel/
        delete-note"
        title="410000018005410c"
        href="http://localhost:8080/dctm-rest/repositories/
            REPO/work-items/4a0000018002900/
            packages/0/notes/410000018005410c"/>
    <link rel="http://identifiers.emc.com/linkrel/work-item"
        href="http://localhost:8080/dctm-rest/repositories/
            REPO/work-items/4a0000018002900"/>
</links>
</package>
```



### Example 2-308: JSON response

```
{
    "package-name": "DocumentToReview",
    "status": "FULL",
    "optional": false,
    "visible": false,
    "operation": "",
    "required-type": "dm_document",
    "note-count": 1,
    "notes": [
        {
            "id": "410000018005410c",
            "text": "this is the first note. true-persistent",
            "creation-date": "2019-08-21T06:37:54.000+00:00",
            "flag": 1,
            "persistent": true,
            "writer": "dave",
            "is-new": false
        }
    ],
    "document-count": 1,
    "document-ids": [
        "0900000180011fad"
    ],
    "skill-level": 0,
    "form-template-defined": true,
    "form-for-properties": false,
    "task-id": "4a0000018002900",
    "index": 0,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/work-items/
                4a0000018002900/packages/0"
        }, {
            "rel": "http://identifiers.emc.com/linkrel/
                delete-documents",
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/work-items/
                4a0000018002900/packages/0/documents"
        }, {
            "rel": "http://identifiers.emc.com/linkrel/document",
            "title": "0900000180011fad",
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/objects/0900000180011fad"
        }, {
            "rel": "http://identifiers.emc.com/linkrel/
                delete-note",
            "title": "410000018005410c",
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/work-items/
```

```

        4a00000180002900/packages/0/notes/
        410000018005410c"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/
            work-item",
        "href": "http://localhost:8080/dctm-rest/
            repositories/REPO/work-items/
            4a00000180002900"
    }
}

```



### Example 2-309: HAL + JSON response

```

{
    "package-name": "DocumentToReview",
    "status": "FULL",
    "optional": false,
    "visible": false,
    "operation": "",
    "required-type": "dm_document",
    "note-count": 1,
    "notes": [
        {
            "id": "410000018005410c",
            "text": "this is the first note. true-persistent",
            "creation-date": "2019-08-21T06:37:54.000+00:00",
            "flag": 1,
            "persistent": true,
            "writer": "dave",
            "is-new": false
        }
    ],
    "document-count": 1,
    "document-ids": [
        "0900000180011fad"
    ],
    "skill-level": 0,
    "form-template-defined": true,
    "form-for-properties": false,
    "task-id": "4a00000180002900",
    "index": 0,
    "_links": {
        "http://identifiers.emc.com/linkrel/document": {
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/objects/
                0900000180011fad",
            "title": "0900000180011fad"
        },
        "http://identifiers.emc.com/linkrel/delete-note": {
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/work-items/
                4a00000180002900/packages/0/notes/
                410000018005410c",
            "title": "410000018005410c"
        },
        "http://identifiers.emc.com/linkrel/work-item": {
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/work-items/
                4a00000180002900"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/
                repositories/REPO/work-items/
                4a00000180002900/packages/0"
        },
        "http://identifiers.emc.com/linkrel/delete-documents": "

```

```
{
    "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/work-items/
        4a00000180002900/packages/0/
        documents"
}
}
```

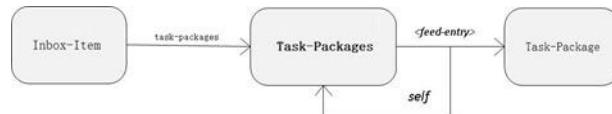


## 2.72.2 Task Packages

The *Task Packages* resource is used to get the packages, otherwise known as attachments, of a specified workflow task.

### 2.72.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Inbox Item” on page 343 and About the Diagram on page 13.

### 2.72.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Workflow task packages
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Task Package” on page 708
- Entry ID: The entry URI
- Entry Title: The name of the package
- Entry Updated: Server's current time

### 2.72.2.3 Link relations

Link relation	Description	Resource reference
self	Link to itself	"Task Packages" on page 719

### 2.72.2.4 Operations

#### 2.72.2.4.1 Supported HTTP methods

Method	Description
GET	Get the workflow task attachment packages.

#### 2.72.2.4.2 Get Workflow Task Packages

Retrieves all packages of a workflow task.

##### 2.72.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	inline				



**Note:** Please note that *page*, *items-per-page*, *include-total*, *sort*, *view*, and *filter* parameters are not supported.

##### 2.72.2.4.2.2 Request media types

N/A

##### 2.72.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.72.2.4.2.4 Request body**

N/A

**2.72.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.72.2.4.2.6 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

**2.72.2.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.72.2.4.2.8 Response body**

 **Example 2-310: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/work-items/
    1b000001800011fb/packages</id>
  <title>Workflow task packages</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2019-02-28T06:35:27.756+00:00</updated>
  <link
    href="http://localhost:8080/dctm-rest/repositories/REPO/work-items/
      1b000001800011fb/packages" rel="self"/>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/work-items/
      1b000001800011fb/packages/0</id>
    <title>Package1</title>
    <summary>EMPTY</summary>
    <updated>2019-02-28T06:35:27.759+00:00</updated>
    <published>2019-02-28T06:35:27.759+00:00</published>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/work-items/
        1b000001800011fb/packages/0" rel="edit"/>
    <content
      src="http://localhost:8080/dctm-rest/repositories/REPO/work-items/0"
```

```

        1b000001800011fb/packages/0" type="application/vnd.emc.documentum+xml"/>
    </entry>
</feed>

```



### Example 2-311: JSON response

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
          1b000001800011fb/packages",
    "title": "Workflow task packages",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-28T06:35:56.803+00:00",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
                  1b000001800011fb/packages"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
                  1b000001800011fb/packages/0",
            "title": "Package1",
            "summary": "EMPTY",
            "updated": "2019-02-28T06:35:56.803+00:00",
            "published": "2019-02-28T06:35:56.803+00:00",
            "links": [
                {
                    "rel": "edit",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                          work-items/1b000001800011fb/packages/0"
                }
            ],
            "content": {
                "type": "application/vnd.emc.documentum+json",
                "src": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
                      1b000001800011fb/packages/0"
            }
        }
    ]
}

```



### Example 2-312: HAL+JSON response

```

{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
          1b000001800011fb/packages",
    "title": "Workflow task packages",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-28T06:35:46.191+00:00",
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
                      1b000001800011fb/packages/0",
                "title": "Package1",
                "summary": "EMPTY",
                "updated": "2019-02-28T06:35:46.191+00:00",
                "published": "2019-02-28T06:35:46.191+00:00",
            }
        ]
    }
}

```

```

        "_links": {
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
work-items/1b000001800011fb/packages/0"
            }
        }
    ],
    "_links": {
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
1b000001800011fb/packages"
        }
    }
}

```



## 2.73 Task Package Documents

The *Task Package Documents* resource is used to remove all the documents in a package.

### 2.73.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Task Package” on page 708 and [About the Diagram](#) on page 13.

### 2.73.2 Feed

No feed support.

### 2.73.3 Link relations

N/A

## 2.73.4 Operations

### 2.73.4.1 Supported HTTP methods

Method	Description
DELETE	Removes all of the documents in a package

### 2.73.4.2 Remove all the documents in the package

Remove all of the documents from a package.

#### 2.73.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE		For more information, see <a href="#">Appendix D, REST Common Definition - HTTP Headers on page 1055</a> .			

#### 2.73.4.2.2 Request media types

N/A

#### 2.73.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.73.4.2.4 Request body

N/A

#### 2.73.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.73.4.2.6 Response media types

N/A

#### 2.73.4.2.7 Response status

- 204 - Removed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.73.4.2.8 Response body

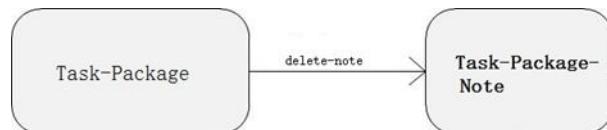
N/A

## 2.74 Task Package Note

The *Task Package Note* resource represents a note in a user's inbox is used to remove the specified note in a Workflow Task Package.

### 2.74.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Task Package](#)” on page 708 and [About the Diagram](#) on page 13.

### 2.74.2 Feed

No feed support.

### 2.74.3 Link relations

N/A

## 2.74.4 Operations

### 2.74.4.1 Supported HTTP methods

Method	Description
DELETE	Remove a specified <i>Package Note</i> .

### 2.74.4.2 Remove specified Package Note

Remove a specified *Package Note*.

#### 2.74.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE		For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.74.4.2.2 Request media types

N/A

#### 2.74.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### 2.74.4.2.4 Request body

N/A

#### 2.74.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.74.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.74.4.2.7 Response status

- 204 - Successfully deleted
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.74.4.2.8 Response body

N/A

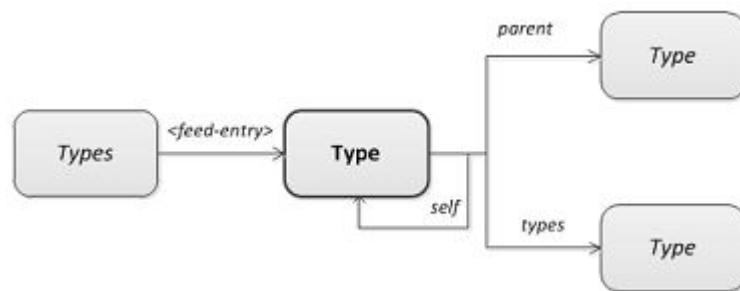
## 2.75 Type (s)

### 2.75.1 Type

The Type resource represents a single type object.

#### 2.75.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Types](#)” on page 755 and [About the Diagram](#) on page 13.

### 2.75.1.2 Feed

No feed support.

### 2.75.1.3 Link relations

Link relation	Description	Resource reference
self	This Type resource.	<a href="#">"Type" on page 727</a>
types [1]	Sub types of this Type object.	<a href="#">"Type" on page 727</a>
parent	Link to the associated Type object.	<a href="#">"Type" on page 727</a>
lightweight-types [1]	The lightweight types of current Shareable Type.	<a href="#">"Types" on page 755</a>
parent-shareable-type [1]	The parent shareable Type of current lightweight Type.	<a href="#">"Type" on page 727</a>
assist-values [1]	The Value Assistance of the Type.	<a href="#">"Value Assists" on page 802</a>
lifecycle [1]	The Lifecycle that is attached.	<a href="#">"Lifecycle" on page 370</a>
attachable-lifecycles [1]	The Lifecycles that can be attached to the current type.	<a href="#">"Lifecycle" on page 370</a>
registered-audit-events [1]	The Registered Audit Events of the current type	<a href="#">"Available Audit Events" on page 106</a>
	 <b>Note:</b> You must have the required privileges to access these events.	
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.75.1.4 Operations

#### 2.75.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a Type object in the repository

#### 2.75.1.4.2 Get a Type

Retrieves the metadata and content properties of the current type object.

##### 2.75.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	inherited	<p>Specifies whether or not to return inherited properties from the parent type.</p> <ul style="list-style-type: none"> <li>• <i>true</i> - Returns inherited properties from the parent type.</li> <li>• <i>false</i> - Only returns properties that directly belong to this type object.</li> </ul>	boolean	<i>true, false</i>	<i>true</i>
	locale	<p>Specifies the locale of the Request.</p> <p> <b>Note:</b> If the corresponding label of the specified locale cannot be found, the label is null.</p>	string	Valid locale	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	include-value-assist	Specifies whether the Response includes the definition information of the value assist	string	<i>true, false</i>	<i>false</i>
	include-lifecycle	Specifies whether to get the type and properties related information for the different lifecycles and states	string	<i>true, false</i>	<i>false</i>
	include-auditable-events	Specifies whether to get the type auditable system and app events	string	<i>true, false</i>	<i>false</i>
	include-display-configs	Specifies whether to get the type display configuration information	string	<i>true, false</i>	<i>false</i>
	include-mapping-tables	Specifies whether to get the type and properties mapping tables information	string	<i>true, false</i>	<i>false</i>
	include-value-constraints	Specifies whether to get the type and properties value constraints information	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	include-all	<p>Specifies whether to get the value assist, lifecycle based type information and auditable events.</p> <p>When either &lt;include-all&gt; or &lt;include-value-assist&gt; are <i>true</i>, the value assist information is returned.</p> <p>When either &lt;include-all&gt; or &lt;include-lifecycle&gt; are <i>true</i>, the lifecycles based type information is returned.</p> <p>When either &lt;include-all&gt; or &lt;include-auditable-events&gt; are <i>true</i>, the type auditable events are returned.</p> <p>When either &lt;include-all&gt; or &lt;include-display-configs&gt; are <i>true</i>, the type display configuration information is returned.</p> <p>When either &lt;include-all&gt;</p>	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
		<p>or &lt;include-mapping-tables&gt; are <i>true</i>, the type and properties mapping table information is returned.</p> <p>When either &lt;include-all&gt; or &lt;include-value-constraints&gt; are <i>true</i>, the type and properties value constraints information is returned.</p>			
scope-config	Specifies whether to get type properties based on the specified scope and display configuration . The value you can set is the scope configuration name. The scope-configuration and the display-configuration must be provided together.	string	scope config name	null	

HTTP method	Query parameters	Description	Data type	Value range	Default value
	display-config Specifies whether to get type properties based on the specified scope and display configuration . The value you can set is the display configuration display name. The scope-configuration and display-configuration must be provided together.	string	display config name	null	



**Note:** When the relative label value of the specified locale is invalid, locale is not returned. Nevertheless, all other attributes are returned.

#### 2.75.1.4.2.2 Request media types

N/A

#### 2.75.1.4.2.3 Request headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.75.1.4.2.4 Request body

N/A

#### 2.75.1.4.2.5 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.75.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.75.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.75.1.4.2.8 Response body

##### ➡ Example 2-313: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<type
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:dm="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    name="my_document" label="My Document" category="standard"
    default-lifecycle="a_default.lifecycle" default-lifecycle-version="CURRENT"

    ignore-parent-constraints="false"
    parent="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject.xml">
    <help-text>xxx</help-text>
    <comment-text>xxx</comment-text>
    <auditable-system-events>
        <event>dm_unaudit</event>
        <event>dm_audit</event>
        ...
    </auditable-system-events>
    <auditable-app-events>
        <event>my_app_event2</event>
        <event>my_app_event1</event>
        ...
    </auditable-app-events>
    <mapping-table>
        <mapping>
            <value>1</value>
            <display>one</display>
        </mapping>
        <mapping>
            <value>2</value>
            <display>two</display>
            <description>2 to two</description>
        </mapping>
    </mapping-table>

    <constraints>
        <constraint>
            <expression>a_property<10000</expression>
            <enforce>BY_APPLICATION</enforce>
            <dependency>a_property</dependency>
            <message>a_property must be less than 10000</message>
        </constraint>
        <constraint>
```

```

<expression>Len(b_property)&lt;100</expression>
<enforce>DISABLED</enforce>
<dependency>b_property</dependency>
<message>b_property must have less than 100 characters</message>
</constraint>
</constraints>
<scope-configs>
  <scope-config>
    <id>6c0000058000050a</id>
    <scope>
      <item>webtop</item>
    </scope>
    <category>
      <item>application</item>
    </category>
    <display-configs>
      <display-config>
        <name>config1</name>
        <id>6b0000058000050a</id>
        <attribute-source>dm_document</attribute-source>
        <fixed-display>false</fixed-display>
        <attribute-hints>
          <attribute-hint>
            <attribute>int_attr1</attribute>
            <display-hint>1</display-hint>
          </attribute-hint>
        </attribute-hints>
      </display-config>
    </display-configs>
  </scope-config>
</scope-configs>
<properties>
  <property name="a_property" category="category_a_property" repeating="false"
    type="string" length="32" label="Type" hidden="false"
    required="false" notnull="false" notnull-enforce="BY_APPLICATION"
    notnull-message="a_property is not nullable" readonly="true"
    searchable="true" display-hint="0" ignore-immutable="true"
    ignore-parent-constraints="true">
    <dependencies>
      <item>attr1</item>
      <item>attr2</item>
    <dependencies>
    <value-assist>
      <query allow-caching="false" allow-user-values="false"
        condition="..." query-attribute="..."></query>
      <query allow-caching="false" allow-user-values="true"
        query-attribute="...">...</query>
    </value-assist>
    <mapping-tables>
      <mapping-table>
        <value>10</value>
        <display>Ten</display>
        <description>10 to Ten</description>
      </mapping-table>
      <mapping-table>
        <value>20</value>
        <display>Twenty</display>
        <description>20 to Twenty</description>
      </mapping-table>
    </mapping-tables>
    <constraints>
      <constraint>
        <expression>a_property&gt;10</expression>
        <enforce>BY_APPLICATION</enforce>
        <dependency>a_property</dependency>
        <message>a_property is larger than 10</message>
      </constraint>
      <constraint>
        <expression>a_property&lt;1000</expression>
        <enforce>BY_APPLICATION</enforce>
      </constraint>
    </constraints>
  </property>
</properties>

```

```
<dependency>a_property</dependency>
<message>a_property is less than 1000</message>
</constraint>
</constraints>
<help-text>The help of the attr a_property</help-text>
<comment-text>The comment of the attr a_property</comment-text>
<lifecycles>
    <lifecycle id="4600000580004914">
        <state name="state1">
            <property-of-lifecycle hidden="false" label="..." notnull="false" notnull-enforce="BY_APPLICATION" notnull-message="a_property is not nullable" readonly="false" required="false" searchable="true" ignore-immutable="true" ignore-parent-constraints="true">
                <value-assist>
                    <fixed-list allow-user-values="true">
                        <value>...</value>
                    </fixed-list>
                </value-assist>
                <constraints>
                    ...
                </constraints>
            </property-of-lifecycle>
        </state>
        <state name="state2">
            <property-of-lifecycle hidden="false" label="..." notnull="false" notnull-enforce="BY_APPLICATION" notnull-message="a_property is not nullable" readonly="false" required="false" searchable="true" ignore-immutable="true" ignore-parent-constraints="true">
                <dependencies>
                    <item>...</item>
                </dependencies>
                <value-assist>
                    <fixed-list allow-user-values="false" condition="...">
                        <value>...</value>
                    </fixed-list>
                    <fixed-list allow-user-values="false" condition="...">
                        <value>...</value>
                    </fixed-list>
                    <fixed-list allow-user-values="false">
                        <value>...</value>
                    </fixed-list>
                </value-assist>
                <constraints>
                    ...
                </constraints>
            </property-of-lifecycle>
        </state>
    </lifecycle>
</properties>
<lifecycles>
    <lifecycle id="4600000580004914">
        <state name="state1">
            <type-of-lifecycle default-lifecycle="xx" default-lifecycle-version="xx" label="xx" />
        </state>
        <state name="state2">
            <type-of-lifecycle default-lifecycle="yy" default-lifecycle-version="yy" label="yy" />
        </state>
    </lifecycle>
</lifecycles>
<links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document.xml"/>
```

```

<link rel="http://identifiers.emc.com/linkrel/types"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
            parent-type=dm_document"/>
<link rel="parent"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types/
            dm_sysobject.xml"/>
<link rel="http://identifiers.emc.com/linkrel/assist-values"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types/
            dm_document/
            assist-values"/>
<link rel="http://identifiers.emc.com/linkrel/lifecycle"
      title="xxx"
      href="xxx"/>
</links>
</type>
```



#### ➤ Example 2-314: JSON response

```
{
  "name": "my_document",
  "label": "My Document",
  "category": "standard",
  "default-lifecycle": "a_default.lifecycle",
  "default-lifecycle-version": "CURRENT",
  "parent": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject",
  "help-text": "xxx",
  "comment-text": "xxx",
  "ignore-parent-constraints": false,
  "auditable-system-events": [
    "dm_unaudit",
    "dm_audit",
    ...
  ],
  "auditable-app-events": [
    "my_app_event1",
    "my_app_event2",
    ...
  ],
  "mapping-table": [
    {
      "value": "1",
      "display": "one"
    },
    {
      "value": "2",
      "display": "two",
      "description": "2 to two"
    }
  ],
  "constraints": [
    {
      "expression": "a_property<10000",
      "enforce": "BY_APPLICATION",
      "dependency": "a_property",
      "message": "a_property must be less than 10000"
    },
    {
      "expression": "Len(b_property) < 100",
      "enforce": "DISABLED",
      "dependency": "b_property",
      "message": "b_property must have less than 100 characters"
    }
  ],
  "scope-configs": [
    {
      "id": "6c0000058000050a",
      "scope": [ "webtop" ],
      "category": [ "application" ],
      ...
    }
  ]
}
```

```
"display-configs" : [ {
    "name" : "config1",
    "id" : "6b0000058000050a",
    "attribute-source" : "dm_document",
    "fixed-display" : false,
    "attribute-hints" : [ {
        "attribute" : "int_attr1",
        "display-hint" : 1
    }]
},
{
    "name": "object_name",
    "repeating": false,
    "type": "string",
    "length": 255,
    "label": "Name",
    "hidden": false,
    "required": false,
    "notnull": true,
    "notnull-enforce" : "BY_APPLICATION",
    "notnull-message" : "a_property is not null",
    "readonly": false,
    "searchable": true,
    "display-hint" : 0,
    "help-text" : "The help of the attr a_property",
    "comment-text" : "The comment of the attr a_property",
    "category" : "category a_property",
    "ignore-immutable" : true,
    "ignore-parent-constraints" : true,
    "dependencies": ["attr1", "attr2"],
    "mapping-tables" : [
        {
            "value" : "10",
            "display" : "Ten",
            "description" : "10 to Ten"
        },
        {
            "value" : "20",
            "display" : "Twenty",
            "description" : "20 to Twenty"
        }
    ],
    "constraints" : [
        {
            "expression" : "a_property>10",
            "enforce" : "BY_APPLICATION",
            "dependency" : "a_property",
            "message" : "a_property is larger than 10"
        },
        {
            "expression" : "a_property<1000",
            "enforce" : "DISABLED",
            "dependency" : "a_property",
            "message" : "a_property is less than 1000"
        }
    ],
    "value-assist" : [
        {
            "type" : "query",
            "condition" : "...",
            "query-attribute" : "...",
            "query-expression" : "...",
            "allow-caching" : false,
            "allow-user-values" : false
        },
        {
            "type" : "query",
```

```

        "query-attribute" : "...",
        "query-expression" : "...",
        "allow-caching" : false,
        "allow-user-values" : true
    }
},
"lifecycles" : {
    "4600000580004914" : {
        "state1" : {
            "label" : "...",
            "hidden" : false,
            "required" : false,
            "notnull" : false,
            "notnull-enforce" : "BY_APPLICATION",
            "notnull-message" : "a_property is not null",
            "readonly" : false,
            "searchable" : true,

            "ignore-immutable" : true,
            "ignore-parent-constraints" : true,
            "dependencies" : [
                ...
            ],
            "value-assist" : [
                {
                    "type" : "query",
                    "condition" : "...",
                    "query-attribute" : "...",
                    "query-expression" : "...",
                    "allow-caching" : true,
                    "allow-user-values" : false
                },
                {
                    "type" : "query",
                    "query-attribute" : "...",
                    "query-expression" : "...",
                    "allow-caching" : false,
                    "allow-user-values" : true
                }
            ],
            "constraints" : [
                ...
            ],
            "state2" : {
                "label" : "b",
                "hidden" : false,
                "required" : false,
                "notnull" : false,
                "notnull-enforce" : "BY_APPLICATION",
                "notnull-message" : "a_property is not null",
                "readonly" : false,
                "searchable" : true,
                "ignore-immutable" : true,
                "ignore-parent-constraints" : true,
                "dependencies" : [
                    ...
                ],
                "value-assist" : [
                    {
                        "type" : "query",
                        "condition" : "...",
                        "query-attribute" : "...",
                        "query-expression" : "...",
                        "allow-caching" : true,
                        "allow-user-values" : true
                    },
                    {
                        "type" : "fixed-list",
                        "values" : [
...
```
```

```
        "..."
    ],
    "allow-user-values" : false
}
],
"constraints" : [
    ...
]
}
},
{
    "name": "r_object_type",
    "repeating": false,
    "type": "string",
    "length": 32,
    "label": "Type",
    "hidden": false,
    "required": false,
    "notnull": false,
    "readonly": true,
    "searchable": true
},
{
    "name": "title",
    "repeating": false,
    "type": "string",
    "length": 400,
    "label": "Title",
    "hidden": false,
    "required": false,
    "notnull": false,
    "readonly": false,
    "searchable": true
},
...
{
    "name": "i_is_replica",
    "repeating": false,
    "type": "boolean",
    "label": "Is Replica",
    "hidden": true,
    "required": false,
    "notnull": false,
    "readonly": true,
    "searchable": true
},
{
    "name": "i_vstamp",
    "repeating": false,
    "type": "integer",
    "label": "Version Stamp",
    "hidden": true,
    "required": false,
    "notnull": false,
    "readonly": true,
    "searchable": false
}
],
"lifecycles" : {
    "4600000580004914" : {
        "state1" : {
            "label" : "Document",
            "default-lifecycle" : "xx",
            "default-lifecycle-version" : "xx"
        },
        "state2" : {
            "label" : "Document",
            "default-lifecycle" : "xx",
            "default-lifecycle-version" : "xx"
        }
    }
}
```

```

        }
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/types",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?parent-type=dm_document"
        },
        {
            "rel": "parent",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/assist-values",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document/assist-values"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/lifecycle",
            "title": "xxx",
            "href": "xxx"
        }
    ]
}

```



### 2.75.1.5 Elements and attributes

**Tip:** New for the 16.4 release

- <*default-lifecycle*> (type level only): The type default lifecycle id
- <*default-lifecycle-version*> (type level only): The version of the default lifecycle
- <*help-text*> (under both type and property level): The type and properties help text
- <*comment-text*> (under both type and property level): The type and properties comment text
- <*auditable-system-events*> (type level only): The names of the auditable system events
- <*auditable-app-events*> (type level only): The names of the auditable application events
- <*mapping-table*> (under both type and property level): The mapping table of the type and properties
- <*scope-configs*> (type level only): The scope configuration and the display configuration information of the type
- <*lifecycles*> (under both type and property level): The type or property information based on different lifecycle states

- *<display-hint>* (property level only): Display-hint is the related display hint for the property, which is added only when both the *<scope-config>* and *<display-config>* are provided as Request parameters
- *<ignore-parent-constraints>* (under both type and property level): Whether or not to ignore the parent type's constraints
- *<constraints>* (under both type and property level): The value constraints for the type and properties
- *<notnull-enforce>* (property level only): How the *<notnull>* is enforced, can be BY\_APPLICATION or DISABLED
- *<notnull-message>* (property level only): The error message for *<notnull>*
- *<ignore-immutable>* (property level only): Whether or not the property is changeable even if the containing object is immutable
- *<category>* (property level only): The user defined property category

### 2.75.1.6 Examples of response

The following sample show you how to use the elements and attributes of this resource.

#### 2.75.1.6.1 Auditable events

When you use the *<include-auditable-events>* parameter and set its value to *true*, the response will contain both the *<auditable-system-events>* and the *<auditable-app-events>*.

##### Example 2-315: XML response

```
<type>
  ...
  <auditable-system-events>
    <event>dm_all</event>
    <event>dm_checkin</event>
    <event>dm_checkout</event>
    ...
  </auditable-system-events>
  <auditable-app-events>
    <event>my_app_event1</event>
    <event>my_app_event2</event>
    <event>my_app_event3</event>
    ...
  </auditable-app-events>
  ...
</type>
```



##### Example 2-316: JSON response

```
{
  ...
  "auditable-system-events" : [
    "dm_all",
    "dm_checkin",
    "dm_checkout",
```

```

        ],
        "auditable-app-events" : [
            "my_app_event1",
            "my_app_event2",
            "my_app_event3",
            ...
        ],
        ...
    }

```



#### 2.75.1.6.2 Display configurations

When you use the `<include-display-configs>` parameter and set its value to `<true>`, the Response will contain both the scope configuration and the display configuration information.

#### Example 2-317: XML response

```

<type>
    ...
    <scope-configs>
        <scope-config>
            <id>6c00000580000100</id>
            <scope>
                <item>webtop</item>
            </scope>
            <category>
                <item>application</item>
            </category>
            <display-configs>
                <display-config>
                    <id>6b00000580000100</id>
                    <name>dm_info</name>
                    <attribute-source>dm_sysobject</attribute-source>
                    <fixed-display>false</fixed-display>
                    <attribute-hints>
                        <attribute-hint>
                            <attribute>object_name</attribute>
                            <display-hint>0</display-hint>
                        </attribute-hint>
                        ...
                        <attribute-hint>
                            <attribute>r_creation_date</attribute>
                            <display-hint>2</display-hint>
                        </attribute-hint>
                        ...
                    </attribute-hints>
                </display-config>
            </display-configs>
        </scope-config>
        <scope-config>
            <id>6c00000580000101</id>
            <scope>
                <item>DcDesktopClient</item>
            </scope>
            <category>
                <item>application</item>
            </category>
            <display-configs>
                ...
            </display-configs>
        </scope-config>
    ...

```

```
</scope-configs>
...
</type>
```



#### ▶ Example 2-318: JSON response

```
{
  ...
  "scope-configs" : [
    {
      "id" : "6c00000580000100",
      "scope" : [
        "webtop"
      ],
      "category" : [
        "application"
      ],
      "display-configs" : [
        {
          "id" : "6b00000580000100",
          "name" : "dm_info",
          "attribute-source" : "dm_sysobject",
          "fixed-display" : false,
          "attribute-hints" : [
            {
              "attribute" : "object_name",
              "display-hint" : 0
            },
            ...
            {
              "attribute" : "r_creation_date",
              "display-hint" : 2
            },
            ...
          ]
        }
      ]
    },
    {
      "id" : "6c00000580000101",
      "scope" : [
        "DcDesktopClient"
      ],
      "category" : [
        "application"
      ],
      "display-configs" : [
        ...
      ]
    },
    ...
  ]
}
```



### 2.75.1.6.3 Mapping tables

When you use the `<include-mapping-tables>` parameter and set its value to `<true>`, the Response will contain the mapping table information for both the type and the properties.

#### ➤ Example 2-319: XML response

```

<type>
  ...
  <mapping-tables>
    <mapping-table>
      <value>dm_info</value>
      <display>Info</display>
    </mapping-table>
    <mapping-table>
      <value>dm_general</value>
      <display>General</display>
    </mapping-table>
    ...
  </mapping-tables>
  <properties>
    <property name="owner_permit" ...>
      <mapping-tables>
        <mapping-table>
          <value>1</value>
          <display>None</display>
        </mapping-table>
        <mapping-table>
          <value>2</value>
          <display>Browse</display>
        </mapping-table>
        <mapping-table>
          <value>3</value>
          <display>Read</display>
        </mapping-table>
        ...
      </mapping-tables>
    </property>
    ...
  </properties>
  ...
</type>

```



#### ➤ Example 2-320: JSON response

```

{
  ...
  "mapping-tables" : [
    {
      "value" : "dm_info",
      "display" : "Info"
    },
    {
      "value" : "dm_general",
      "display" : "General"
    },
    ...
  ],
  "properties" : [
    {
      "name" : "owner_permit",
      ...
      "mapping-tables" : [

```

```
{  
    "value" : "1",  
    "display" : "None"  
},  
{  
    "value" : "2",  
    "display" : "Browse"  
},  
{  
    "value" : "3",  
    "display" : "Read"  
},  
...  
],  
...  
1,  
...  
}
```



#### 2.75.1.6.4 Value constraints

When you use the `<include-value-constraints>` parameter and set its value to `true`, the Response will contain constraints for both the type and properties.



#### Example 2-321: XML response

```
<type>  
    ...  
    <constraints>  
        <constraint>  
            <expression>property_c<10000</expression>  
            <enforce>BY_APPLICATION</enforce>  
            <dependency>property_c</dependency>  
            <message>property_c must be less than 10000</message>  
        </constraint>  
        <constraint>  
            <expression>Len(property_d) < 100</expression>  
            <enforce>DISABLED</enforce>  
            <dependency>property_d</dependency>  
            <message>property_d must have less than 100 characters</message>  
        </constraint>  
        ...  
    </constraints>  
    <properties>  
        <property name="property_a" ...>  
            ...  
            <constraints>  
                <constraint>  
                    <expression>property_a > 10</expression>  
                    <enforce>BY_APPLICATION</enforce>  
                    <dependency>property_a</dependency>  
                    <message>property_a is larger than 10</message>  
                </constraint>  
                <constraint>  
                    <expression>property_a<1000</expression>  
                    <enforce>BY_APPLICATION</enforce>  
                    <dependency>property_a</dependency>  
                    <message>property_a is less than 1000</message>  
                </constraint>  
            </constraints>  
        </property>  
        ...  
    </properties>
```

```
</type>
...

```



### Example 2-322: JSON response

```
{
  ...
  "constraints" : [
    {
      "expression" : "property_c<10000",
      "enforce" : "BY_APPLICATION",
      "dependency" : "property_c",
      "message" : "property_c must be less than 10000"
    },
    {
      "expression" : "Len(property_d) < 100",
      "enforce" : "DISABLED",
      "dependency" : "property_d",
      "message" : "property_d must have less than 100 characters"
    }
    ...
  ],
  "properties" : [
    {
      "name" : "property_a",
      ...
      "constraints" : [
        {
          "expression" : "property_a > 10",
          "enforce" : "BY_APPLICATION",
          "dependency" : "property_a",
          "message" : "property_a is larger than 10"
        },
        {
          "expression" : "property_a<1000",
          "enforce" : "BY_APPLICATION",
          "dependency" : "property_a",
          "message" : "property_a is less than 1000"
        }
      ]
    },
    ...
  ],
  ...
}
```



#### 2.75.1.6.5 Value assistances

When you use the `<include-value-assist>` parameter and set its value to `true`, the Response will contain the value assistance definition.



**Note:** Dependencies are returned as basic property information.

### Example 2-323: XML response

```
<type>
  ...
  <properties>
    <property name="property_a">
```

```

...
<value-assist>
    <fixed-list allow-user-values="false">
        <value>1</value>
        <value>2</value>
        <value>3</value>
    </fixed-list>
</value-assist>
</property>
<property name="property_b">
    ...
    <dependencies>
        <item>property_b</item>
    </dependencies>
    <value-assist>
        <query allow-caching="false" allow-user-values="false"
            condition="property_a=1"
            query-attribute="user_name">select user_name from dm_user</query>
        <query allow-caching="false" allow-user-values="true"
            condition="property_a=2"
            query-attribute="name">select name from dm_format</query>
        <fixed-list allow-user-values="false">
            <value>b</value>
            <value>bb</value>
            <value>bbb</value>
        </fixed-list>
    </value-assist>
</property>
    ...
</properties>
    ...
</type>

```



#### ➡ Example 2-324: JSON response

```
{
...
"properties" : [
    {
        "name" : "property_a",
        ...
        "value-assist" : [
            {
                "type" : "fixed-list",
                "values" : [
                    "1",
                    "2",
                    "3"
                ],
                "allow-user-values" : false
            }
        ]
    },
    {
        "name" : "property_b",
        ...
        "dependencies" : [
            "property_aa"
        ],
        "value-assist" : [
            {
                "type" : "query",
                "condition" : "property_a=1",
                "query-attribute" : "user_name",
                "query-expression" : "select user_name from dm_user",
                "allow-caching" : false,
            }
        ]
    }
]
```

```

        "allow-user-values" : false
    },
{
    "type" : "query",
    "condition" : "property_a=2",
    "query-attribute" : "name",
    "query-expression" : "select name from dm_format",
    "allow-caching" : false,
    "allow-user-values" : true
},
{
    "type" : "fixed-list",
    "values" : [
        "b",
        "bb",
        "bbb"
    ],
    "allow-user-values" : false
}
],
...
],
...
}

```



#### 2.75.1.6.6 Definition of lifecycle states

When you use the `<include-lifecycle>` parameter and set its value to `true`, the Response contains different lifecycle values according to individual lifecycle states.

Currently, the server supports the information retrieval for the following Lifecycle states:

##### Type

- `<default-lifecycle>`
- `<default-lifecycle-version>`
- `<ignore-parent-constraints>`
- `<constraints>`

##### Property

- `<label>`
- `<hidden>`
- `<required>`
- `<notnull>`
- `<notnull-enforce>`
- `<notnull-message>`
- `<readonly>`
- `<searchable>`

- <ignore-parent-constraints>
- <ignore-immutable>
- <dependencies>
- <value-assist>
- <constraints>

Only those values that are different than the <type> or <property> without the Lifecycle are returned. Duplicate values are not returned.

In addition to the <include-lifecycle> parameter, other 'include-xxx' type parameters are required when you want to get the value of each Lifecycle state.

For example, when:

- include-lifecycle=true and include-value-assist=true, this returns the value assist definition of the properties. In addition to this, when value assist definitions exists for the different lifecycle states, then they are also returned.

#### ➤ Example 2-325: XML response

```
<type>
  ...
  <properties>
    <property name="property_a" ...>
      ...
      <lifecycles>
        <lifecycle id="4600000580006b20">
          <state name="state_1">
            <property-of-lifecycle>
              <dependencies>
                <item>property_b</item>
              </dependencies>
              <value-assist>
                <fixed-list allow-user-values="false"
condition="property_b>1">
                  <value>1000</value>
                </fixed-list>
                <fixed-list allow-user-values="false">
                  <value>9999</value>
                </fixed-list>
              </value-assist>
            </property-of-lifecycle>
          </state>
          <state name="state_2">
            <property-of-lifecycle>
              ...
              </property-of-lifecycle>
            </state>
          </lifecycle>
        <lifecycle id="4600000580006b21">
          ...
          <lifecycle>
        </lifecycles>
      </property>
      ...
    </properties>
    <lifecycles>
      <lifecycle id="4600000580006b20">
        <state name="state_1">
          <type-of-lifecycle default-lifecycle="4600000580006b20">
```

```

        default-lifecycle-version="CURRENT" />
    </state>
    <state name="state_2">
        <type-of-lifecycle default-lifecycle="4600000580006b21"
                           default-lifecycle-version="CURRENT" />
    </state>
</lifecycle>
</lifecycles>
...
</type>

```



### Example 2-326: JSON response

```

{
...
"properties" : [
    {
        "name" : "property_a",
        ...
        "lifecycles" : {
            "4600000580006b20" : {
                "state_1" : {
                    "dependencies" : [
                        "property_b"
                    ],
                    "value-assist" : [
                        {
                            "type" : "fixed-list",
                            "condition" : "property_b>1",
                            "values" : [
                                "1000"
                            ],
                            "allow-user-values" : false
                        },
                        {
                            "type" : "fixed-list",
                            "values" : [
                                "9999"
                            ],
                            "allow-user-values" : false
                        }
                    ]
                },
                "DzCzCXAizTwKyP8_0" : {
                    ...
                }
            },
            "4600000580006b21" : {
                ...
            }
        }
    },
    ...
]
"lifecycles" : {
    "4600000580006b20" : {
        "state_1" : {
            "default-lifecycle" : "4600000580006b20",
            "default-lifecycle-version" : "CURRENT"
        },
        "state_2" : {
            "default-lifecycle" : "4600000580006b21",
            "default-lifecycle-version" : "CURRENT"
        }
    }
},
...
}

```

```
 } ...
```



#### 2.75.1.6.7 Include all

When you want to retrieve all of the '*<include-xxx>*' type values in the response, use the *<include-all>* parameter and set its value to *true*.

#### 2.75.1.6.8 Get the properties of scope-config or display-config

When you use the *<scope-config>* and *<display-config>* parameters, then only those properties that are defined in the related display config are returned. These two parameters must be provided together. The *<display-hint>* value is returned in addition to the normal information that is returned.

Other '*<include-xxx>*' type parameters may be used as well.



#### Example 2-327: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<type>
  ...
  <properties>
    <property display-hint="0" name="object_name" .../>
    <property display-hint="0" name="title" .../>
    ...
    <property display-hint="2" name="log_entry" .../>
    ...
  </properties>
  ...
</type>
```



#### Example 2-328: JSON response

```
{
  "name" : "dm_document",
  ...
  "properties" : [
    {
      "name" : "object_name",
      ...
      "display-hint" : 0
    },
    {
      "name" : "title",
      ...
      "display-hint" : 0
    },
    ...
    {
      "name" : "log_entry",
      "display-hint" : 2
    },
    ...
  ],
  ...
}
```



### 2.75.1.7 Example of request and response

#### ➤ Example 2-329: XML request

```
GET http://localhost:8080/dctm-rest/repositories/acme/types/dm_document
Accept:application/vnd.emc.documentum+xml
```



#### ➤ Example 2-330: XML response

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8
<?xml version="1.0" encoding="UTF-8"?>
<type category="standard" ignore-parent-constraints="false"
      label="Document" name="dm_document"
      parent="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject"
      xmlns="http://identifiers.emc.com/vocab/documentum"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <properties>
    <property hidden="false" ignore-immutable="false"
              ignore-parent-constraints="false" label="Name" length="255"
              name="object_name" notnull="false" readonly="false"
              repeating="false" required="false" searchable="true" type="string"/>
    <property hidden="false" ignore-immutable="false"
              ignore-parent-constraints="false" label="Type" length="32"
              name="r_object_type" notnull="false" readonly="true"
              repeating="false" required="false" searchable="true" type="string"/>
    ...
  </properties>
  <links>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document"
      rel="self"/>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/types?parent-
            type=dm_document"
      rel="http://identifiers.emc.com/linkrel/types"/>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject"
      rel="parent"/>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document/
            assist-values"
      rel="http://identifiers.emc.com/linkrel/assist-values"/>
  </links>
</type>
```



#### ➤ Example 2-331: JSON request

```
GET http://localhost:8080/dctm-rest/repositories/acme/types/dm_document
Accept:application/vnd.emc.documentum+json
```



 Example 2-332: JSON response

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "name" : "dm_document",
  "label" : "Document",
  "category" : "standard",
  "parent" : "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject",
  "ignore-parent-constraints" : false,
  "properties" : [
    {
      "name" : "object_name",
      "repeating" : false,
      "type" : "string",
      "length" : 255,
      "label" : "Name",
      "hidden" : false,
      "required" : false,
      "notnull" : false,
      "readonly" : false,
      "searchable" : true,
      "ignore-immutable" : false,
      "ignore-parent-constraints" : false
    },
    {
      "name" : "r_object_type",
      "repeating" : false,
      "type" : "string",
      "length" : 32,
      "label" : "Type",
      "hidden" : false,
      "required" : false,
      "notnull" : false,
      "readonly" : true,
      "searchable" : true,
      "ignore-immutable" : false,
      "ignore-parent-constraints" : false
    },
    ...
  ],
  "links" : [
    {
      "rel" : "self",
      "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document"
    },
    {
      "rel" : "http://identifiers.emc.com/linkrel/types",
      "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types?parent-type=dm_document"
    },
    {
      "rel" : "parent",
      "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject"
    },
    {
      "rel" : "http://identifiers.emc.com/linkrel/assist-values",
      "href" : "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document/assist-values"
    }
  ]
}
```

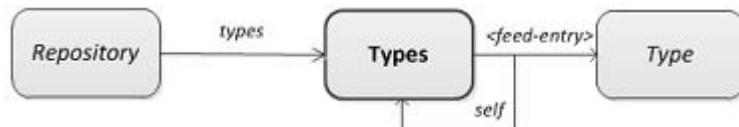


## 2.75.2 Types

The Types collection resource represents the collection of type objects in a repository.

### 2.75.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Repository” on page 564 and About the Diagram on page 13.

### 2.75.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Types collection
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Type” on page 727
- Entry ID: Type URI
- Entry Title: Display name of the type.
- Entry Updated: r\_modify\_date of the object

### 2.75.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of type objects	“Types” on page 755

### 2.75.2.4 Operations

#### 2.75.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of types from the repository.

#### 2.75.2.4.2 Get Types

Retrieve a collection of types from the repository.

##### 2.75.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	parent-type	<p>Gets direct sub types or all sub types of the specified type.</p> <p>You can combine this variable with recursive to determine whether or not to return all sub types or just the direct sub types of the specified type.</p>	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	dml-view	<p>Specifies what shall be returned for the collection. Valid only when <code>inline=true</code>:</p> <ul style="list-style-type: none"> <li>• <code>dml-view=all</code>: Returns all type and types properties attributes, including: <ul style="list-style-type: none"> <li>– name, label, and parent</li> <li>– All properties of the properties in the type, including: name, label, type, length, repeating, hidden, notnull, readonly, required, default, and searchable</li> </ul> </li> <li>• <code>dml-view=basic</code>: Returns only basic dml, including: <ul style="list-style-type: none"> <li>– The basic properties of the type: name, and parent</li> <li>– The basic properties of the properties in the type, including: name, type, length, and repeating</li> </ul> </li> <li>• <code>dml-view=none</code>: Returns only a minimum set of the content, including the name, and the parent of the type. Type label and all information about the properties in the type are not returned.</li> </ul>	string	N/A	basic
	inherited	Specifies whether or not to include all of the properties of a type including the inherited properties from the parent type. This parameter does not work when <code>dml-view=none</code> .	string	<i>true, false</i>	<i>true</i>
	locale	Specifies the locale of the Request. This parameter only works when <code>dml-view=all</code> .	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	include-lifecycle	Specifies whether or not to get the type related information for the different lifecycles and states, the parameter is valid only when <code>dml-view=all</code> .	string	<i>true, false</i>	<i>false</i>
	include-auditable-events	Specifies whether or not to get the type auditable system and app events, the parameter is valid only when <code>dml-view=all</code> .	string	<i>true, false</i>	<i>false</i>
	include-display-configs	Specifies whether or not to get the type display configuration information, the parameter is valid only when <code>dml-view=all</code> .	string	<i>true, false</i>	<i>false</i>
	include-mapping-tables	Specifies whether or not to get the type and properties mapping table information, the parameter is valid only when <code>dml-view=all</code> .	string	<i>true, false</i>	<i>false</i>
	include-value-constraints	Specifies whether or not to get the type and properties value constraints information.	string	<i>true, false</i>	<i>false</i>

HTTP method	Query parameters	Description	Data type	Value range	Default value
	include-all	<p>Specifies whether or not to get both lifecycle based type information, and auditable events. the parameter is valid only when <code>dml-view=all</code>.</p> <p>When one of <code>include-all</code> and <code>include-lifecycle</code> is <i>true</i>, the Lifecycles based type information is returned.</p> <p>When one of <code>include-all</code> and <code>include-auditable-events</code> is <i>true</i>, the type auditable events are returned.</p> <p>When one of <code>include-all</code> and <code>include-display-configs</code> is <i>true</i>, the type display configuration information is returned.</p> <p>When one of <code>include-all</code> and <code>include-mapping-tables</code> is <i>true</i>, the type mapping table information is returned.</p> <p>When one of <code>include-all</code> and <code>include-value-constraints</code> is <i>true</i>, the type and properties value constraints information are returned.</p>	string	<i>true, false</i>	<i>false</i>

This method also supports the following standard query parameters:

- `sort`
- `start-with`
- `inline`
- `recursive`
- `items-per-page`
- `page`
- `include-total`

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).



**Note:** The default sort order is by type name.

The view query parameter is not supported by this resource.

#### 2.75.2.4.2.2 Request media types

N/A

#### 2.75.2.4.2.3 Request headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.75.2.4.2.4 Request body

N/A

#### 2.75.2.4.2.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.75.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.75.2.4.2.7 Response status

- 200 - Retrieved successfully
- 201, 204, 400, 401, 403, 404, 406, 409, 415, 500

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.75.2.4.2.8 Response body

XML or JSON representation of the collection of type objects.



**Note:** The type name in the URL must be encoded.



#### Example 2-333: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<feed
      xmlns="http://www.w3.org/2005/Atom"
      xmlns:dm="http://identifiers.emc.com/vocab/documentum"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```

<id>http://localhost:8080/dctm-rest/repositories/REPO/types</id>
<title>Types</title>
<author>
    <name>OpenText Documentum</name>
</author>
<updated>2015-11-18T02:43:19.185+00:00</updated>
<dm:page>1</dm:page>
<dm:items-per-page>1</dm:items-per-page>
<link rel="self"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?items-per-
page=1" />
<link rel="next"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
      items-per-page=1&page=2" />
<link rel="first"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
      items-per-page=1&page=1" />
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl</id>
    <title>dm_acl</title>
    <author>
        <name>REPO_ADMIN</name>
        <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/REPO_ADMIN.xml</
uri>
    </author>
    <summary>0300000580000101</summary>
    <updated>2015-11-18T02:43:19.185+00:00</updated>
    <published>2015-11-18T02:43:19.185+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl.xml" />
    <content type="application/xml"
              src="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_acl.xml" />
    </entry>
</feed>

```



#### Example 2-334: JSON request

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/types",
    "title": "Types",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2015-11-18T02:42:39.442+00:00",
    "page": 1,
    "items-per-page": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?items-per-
page=1"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?
            items-per-page=1&page=2"
        },
        {
            "rel": "first",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?
            items-per-page=1&page=1"
        }
    ],
}
```

```

    "entries": [
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl",
        "title": "dm_acl",
        "author": [
          {
            "name": "REPO_ADMIN",
            "uri": "http://localhost:8080/dctm-rest/repositories/REPO/users/
REPO_ADMIN"
          }
        ],
        "summary": "0300000580000101",
        "updated": "2015-11-18T02:42:39.443+00:00",
        "published": "2015-11-18T02:42:39.443+00:00",
        "links": [
          {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_acl"
          }
        ],
        "content": {
          "type": "application/vnd.emc.documentum+json",
          "src": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl"
        }
      }
    ]
  }
}

```



**Note:** In release 7.1, the MIME type property was named content-type. This name was deprecated and replaced with MIME type property name type. You can set the rest.api.compatibility.version property in rest-api-runtime.properties to 7.1 for backward compatibility.

#### 2.75.2.4.2.9 Examples of request and response



##### Example 2-335: XML request

```

GET http://localhost:8080/dctm-rest/repositories/REPO/types?
  items-per-page=1&inline=true&dml-view=all
Accept:application/vnd.emc.documentum+xml

```



##### Example 2-336: XML response

```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<?xml version='1.0' encoding='UTF-8'?>
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:dm="http://identifiers.emc.com/vocab/documentum"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/types</id>
  <title>Types</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2015-11-18T02:51:22.186+00:00</updated>
  <dm:page>1</dm:page>

```

```

<dm:items-per-page>1</dm:items-per-page>
<link rel="self"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
             items-per-page=1&amp;inline=true&amp;dml-view=all"/>
<link rel="next"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
             items-per-page=1&amp;inline=true&amp;dml-
view=all&amp;page=2"/>
<link rel="first"
      href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
             items-per-page=1&amp;inline=true&amp;dml-
view=all&amp;page=1"/>
<entry>
  <id>http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl</id>
  <title>dm_acl</title>
  <author>
    <name>REPO_ADMIN</name>
    <uri>http://localhost:8080/dctm-rest/repositories/REPO/users/REPO_ADMIN.xml</uri>
  </author>
  <summary>0300000580000101</summary>
  <updated>2015-11-18T02:51:22.187+00:00</updated>
  <published>2015-11-18T02:51:22.187+00:00</published>
  <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl.xml"/>
  <content>
    <dm:type name="dm_acl" label="ACL" category="standard">
      <dm:properties>
        <property name="object_name" repeating="false" type="string" length="32"
                 label="Permissions Set Name" hidden="false" required="false"
                 notnull="false" readonly="false" searchable="true"/>
        <property name="description" repeating="false" type="string" length="128"
                 label="Permissions Set Description" hidden="false"
                 required="false"
                 notnull="false" readonly="false" searchable="true"/>
        <property name="owner_name" repeating="false" type="string" length="255"
                 label="Permissions Set Owner" hidden="false" required="false"
                 notnull="false" readonly="false" searchable="true"/>
        <property name="r_is_internal" repeating="false" type="boolean" label="Is
Internal"
                  hidden="false" required="false" notnull="false" readonly="true"
                  searchable="false"/>
        <property name="r_accessor_name" repeating="true" type="string" length="255"
                 label="Permissions Set Members" hidden="false" required="false"
                 notnull="false" readonly="true" searchable="true"/>
        <property name="r_accessor_permit" repeating="true" type="integer"
                 label="Assigned Permission" hidden="false" required="false"
                 notnull="false" readonly="true" searchable="true"/>
        <property name="r_accessor_xpermit" repeating="true" type="integer"
                 label="Extended Permissions" hidden="false" required="false"
                 notnull="true" readonly="true" searchable="true"/>
        <property name="r_is_group" repeating="true" type="boolean" label="Is Group"
                 hidden="false" required="false" notnull="false" readonly="true"
                 searchable="true"/>
        <property name="globally_managed" repeating="false" type="boolean"
label="Global"
                  hidden="false" required="false" notnull="false" readonly="false"
                  searchable="true"/>
        <property name="acl_class" repeating="false" type="integer" label="ACL
Class"
                  hidden="false" required="false" notnull="false" readonly="false"
                  searchable="true"/>
        <property name="r_has_events" repeating="false" type="boolean" label="Has
Events"
                  hidden="false" required="false" notnull="false" readonly="true"
                  searchable="true"/>
        <property name="r_permit_type" repeating="true" type="integer" label="Permit
Type"
                  hidden="false" required="false" notnull="false" readonly="true"
                  searchable="true"/>
        <property name="r_application_permit" repeating="true" type="string"
                  hidden="false" required="false" notnull="false" readonly="true"
                  searchable="true"/>
      </dm:properties>
    </dm:type>
  </content>
</entry>

```

```

length="128"
          label="Application Permit" hidden="false" required="false"
          notnull="false" readonly="true" searchable="true"/>
<property name="i_has_required_groups" repeating="false" type="boolean"
          label="Has Yes Group(s)" hidden="true" required="false"
          notnull="false" readonly="true" searchable="true"/>
<property name="i_has_required_group_set" repeating="false" type="boolean"
          label="Has Yes Group Set" hidden="true" required="false"
          notnull="false" readonly="true" searchable="true"/>
<property name="i_has_access_restrictions" repeating="false" type="boolean"
          label="Has Access Restrictions" hidden="true" required="false"
          notnull="false" readonly="true" searchable="true"/>
<property name="r_template_id" repeating="false" type="String" length="16"
          label="Template ID" hidden="false" required="false"
          notnull="false"
          readonly="true" searchable="true"/>
<property name="r_alias_set_id" repeating="false" type="String" length="16"
          label="Alias Set ID" hidden="false" required="false"
          notnull="false"
          readonly="true" searchable="true"/>
<property name="i_partition" repeating="false" type="integer"
          label="Partition Number" hidden="true" required="false"
          notnull="false" readonly="true" searchable="true"/>
<property name="i_is_replica" repeating="false" type="boolean"
          label="Replica"
          hidden="true" required="false" notnull="false" readonly="true"
          searchable="true"/>
<property name="i_vstamp" repeating="false" type="integer" label="Version
Stamp"
          hidden="true" required="false" notnull="false" readonly="true"
          searchable="true"/>
</dm:properties>
<dm:links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_acl.xml"/>
    <link rel="http://identifiers.emc.com/linkrel/types"
        href="http://localhost:8080/dctm-rest/repositories/REPO/types.xml?
parent-type=dm_acl"/>
</dm:links>
</dm:type>
</content>
</entry>
</feed>

```



### Example 2-337: JSON request

```

GET http://localhost:8080/dctm-rest/repositories/REPO/types?
  items-per-page=1&inline=true&dm1-view=all
Accept:application/vnd.emc.documentum+json

```



### Example 2-338: JSON response

```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/types",
  "title": "Types",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ]
}

```

```

        }
    ],
    "updated": "2015-11-18T02:54:09.968+00:00",
    "page": 1,
    "items-per-page": 1,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?
                    items-per-page=1&inline=true&dml-view=none"
        },
        {
            "rel": "next",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?
                    items-per-page=1&inline=true&dml-view=none&page=2"
        },
        {
            "rel": "first",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?
                    items-per-page=1&inline=true&dml-view=none&page=1"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl",
            "title": "dm_acl",
            "author": [
                {
                    "name": "REPO_ADMIN",
                    "uri": "http://localhost:8080/dctm-rest/repositories/REPO/users/REPO_ADMIN"
                }
            ],
            "summary": "0300000580000101",
            "updated": "2015-11-18T02:54:09.969+00:00",
            "published": "2015-11-18T02:54:09.969+00:00",
            "links": [
                {
                    "rel": "edit",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl"
                }
            ],
            "content": {
                "name": "dm_acl",
                "category": "standard",
                "links": [
                    {
                        "rel": "self",
                        "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_acl"
                    },
                    {
                        "rel": "http://identifiers.emc.com/linkrel/types",
                        "href": "http://localhost:8080/dctm-rest/repositories/REPO/types?parent-
type=dm_acl"
                    }
                ]
            }
        }
    ]
}

```



### GET a category of types

The following type of categories are available:

- 0 - standard object type

- 1 - aspect property object type
- 2 - shareable object type
- 4 - lightweight object type
- 8 - data table type

The `type_category` attribute is used along with the `filter` query parameter and the `category` type value to retrieve all of the types of a specified category. Here are some examples that show you how to use the above type category values with the `filter` query parameter. In each example, we use a repository name in the URL, the `filter` query parameter, the `type_category` attribute, and one of the category values (shown above) to retrieve all of the objects of that type.

#### ➤ Example 2-339: GET all lightweight object types

In this example, we use a category value of 4 to get all of the lightweight object types.

```
GET http://localhost:8080/dctm-rest/repositories/{repositoryName}/types?
filter=type_category=4
```



#### ➤ Example 2-340: GET all shareable object types

In this example, we use a category value of 2 to get all of the shareable object types.

```
GET http://localhost:8080/dctm-rest/repositories/{repositoryName}/types?
filter=type_category=2
```



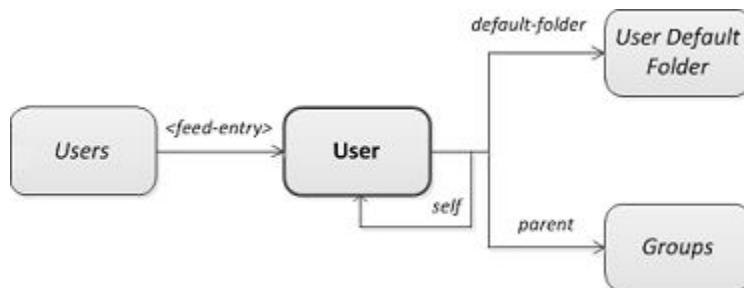
## 2.76 User (s)

### 2.76.1 User

The `User` resource represents the metadata of a user in a repository.

#### 2.76.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Users” on page 775 and About the Diagram on page 13.

### 2.76.1.2 Feed

No feed support.

### 2.76.1.3 Link relations

Link relation	Description	Resource reference
self	This User resource.	<a href="#">“User” on page 766</a>
default-folder [1]	Default folder (home cabinet) of the user.	<a href="#">“User Default Folder” on page 780</a>
parent	Groups that the user belongs to.	<a href="#">“Groups” on page 317</a>
edit	This User resource. Only SysAdmin or SuperUser can view this link	<a href="#">“User” on page 766</a>
delete [1]	This User resource. Only SysAdmin or SuperUser can view this link	<a href="#">“User” on page 766</a>
permission-set [1]	The permission set definition for the User	<a href="#">“Permission Set” on page 507</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

### 2.76.1.4 Operations

#### 2.76.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a User
POST	Update a User
DELETE	Delete a User

#### 2.76.1.4.2 Get a User

Retrieves user details such as title, summary, updated, published, and links details.

The response of the /users/user REST API is modified to enhance the security and share only non-sensitive user details.

By default, the REST API returns the only the following subset of user attributes:

- r\_object\_id
- user\_name
- user\_address
- r\_is\_group

The REST API follows the previously defined behavior if:

- The logged-in user requests the details of their own account.
- The requesting user is an administrator, super user, or a member of the administrators or superusers group.
- The `rest.restrict.user.attributes` property in `rest-api-runtime.properties` is explicitly set to `False`.

#### **2.76.1.4.2.1 HTTP Request method and query parameters**

GET

This method supports the `view` common query parameter.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

#### **2.76.1.4.2.2 Request media types**

N/A

#### **2.76.1.4.2.3 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.76.1.4.2.4 Request body**

N/A

#### **2.76.1.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.76.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.76.1.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.76.1.4.2.8 Response body

 **Note:** The password attribute is not displayed.

##### Example 2-341: XML response

```
<user xsi:type="dm_user" definition="http://localhost:8080/dctm-rest/repositories/REPO/
    types/dm_user.xml">
    <properties>
        <user_name>Administrator</user_name>
        <user_os_name>Administrator</user_os_name>
        <user_address>J0@OpenText.COM</user_address>
        <user_group_name>docu</user_group_name>
        <user_privileges>8</user_privileges>
        <owner_def_permit>7</owner_def_permit>
        <world_def_permit>3</world_def_permit>
        <group_def_permit>5</group_def_permit>
        <default_folder>/Administrator</default_folder>
        <r_is_group>false</r_is_group>
        <user_db_name>Administrator</user_db_name>
        <description />
        <acl_domain>Administrator</acl_domain>
        <acl_name>dn_4500000580000101</acl_name>
        <user_os_domain>restcs72ga</user_os_domain>
        <home_docbase />
        <user_state>0</user_state>
        <client_capability>8</client_capability>
        <globally_managed>false</globally_managed>
        <r_modify_date>2015-03-05T07:55:35.000+00:00</r_modify_date>
        <user_delegation />
        <workflow_disabled>false</workflow_disabled>
        <alias_set_id>0000000000000000</alias_set_id>
        <user_source>inline password</user_source>
        <user_ldap_dn /> //this property is deprecated
        <user_xprivileges>0</user_xprivileges>
        <r_has_events>false</r_has_events>
        <failed_auth_attempt>-1</failed_auth_attempt>
        <user_admin />
        <user_global_unique_id>restcs72ga:Administrator</user_global_unique_id>
        <user_login_name>Administrator</user_login_name>
        <user_login_domain>restcs72ga</user_login_domain>
        <user_initials />
        <user_web_page />
        <first_failed_auth_utc_time xsi:nil="true" />
        <last_login_utc_time>2015-03-02T18:15:33.000+00:00</last_login_utc_time>
```

```

<deactivated_utc_time xsi:nil="true" />
<deactivated_ip_addr />
<restricted_folder_ids xsi:nil="true" />
<root_log_dir />
<i_is_replica>false</i_is_replica>
<i_vstamp>4</i_vstamp>
<r_object_id>1100000580000102</r_object_id>
</properties>
<links>
    <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/users/
        Administrator" />
    <link rel="parent" href="http://localhost:8080/dctm-rest/repositories/REPO/
        groups.xml?user-name=Administrator" />
    <link rel="http://identifiers.emc.com/linkrel/default-folder"
        href="http://localhost:8080/dctm-rest/repositories/REPO/users/
        Administrator/home.xml" />
</links>
</user>

```



### Example 2-342: JSON response

```
{
    "name": "user",
    "type": "dm_user",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_user.json",
    "properties": {
        "user_name": "Administrator",
        "user_os_name": "Administrator",
        "user_address": "J0@OpenText.COM",
        "user_group_name": "docu",
        "user_privileges": 8,
        "owner_def_permit": 7,
        "world_def_permit": 3,
        "group_def_permit": 5,
        "default_folder": "/Administrator",
        "r_is_group": false,
        "user_db_name": "Administrator",
        "description": "",
        "acl_domain": "Administrator",
        "acl_name": "dm_4500000580000101",
        "user_os_domain": "restcs72ga",
        "home_docbase": "",
        "user_state": 0,
        "client_capability": 8,
        "globally_managed": false,
        "r_modify_date": "2015-03-05T07:55:35.000+00:00",
        "user_delegation": "",
        "workflow_disabled": false,
        "alias_set_id": "0000000000000000",
        "user_source": "inline password",
        "user_ldap_dn": "", //this property is deprecated
        "user_xprivileges": 0,
        "r_has_events": false,
        "failed_auth_attempt": -1,
        "user_admin": "",
        "user_global_unique_id": "restcs72ga:Administrator",
        "user_login_name": "Administrator",
        "user_login_domain": "restcs72ga",
        "user_initials": "",
        "user_web_page": "",
        "first_failed_auth_utc_time": null,
        "last_login_utc_time": "2015-03-02T18:15:33.000+00:00",
        "deactivated_utc_time": null,
        "deactivated_ip_addr": "",
        "restricted_folder_ids": null,
        "root_log_dir": ""
    }
}
```

```

        "i_is_replica":false,
        "i_vstamp":4,
        "r_object_id":"1100000580000102"
    },
    "links":[{"rel":"self",
    "href":"http://localhost:8080/dctm-rest/repositories/REPO/users/Administrator.json"},
    {"rel":"parent",
    "href":"http://localhost:8080/dctm-rest/repositories/REPO/groups.json?
    user-name=Administrator"}, {"rel":"http://identifiers.emc.com/linkrel/default-folder",
    "href":"http://localhost:8080/dctm-rest/repositories/REPO/users/
    Administrator/home.json"}]
}

```



#### 2.76.1.4.3 Update a User

Updates the attributes of a user.

##### 2.76.1.4.3.1 HTTP Request method and query parameters

POST

##### 2.76.1.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

##### 2.76.1.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.76.1.4.3.4 Request body

The following read-only properties cannot be updated:

- i\_vstamp
- i\_is\_replica
- r\_has\_events
- r\_modify\_date
- r\_is\_group

 **Example 2-343: XML request**

```
<user xsi:type="dm_user" definition="http://localhost:8080/dctm-rest/repositories/
REPO/types/dm_user.xml">
    <properties>
        <user_address>J0@OpenText.COM</user_address>
        <user_group_name>docu</user_group_name>
        <user_privileges>8</user_privileges>
        .....
    </properties>
</user>
```

 **Example 2-344: JSON request**

```
{
    "properties":{
        "user_address": "J0@OpenText.COM",
        "user_group_name": "docu",
        "user_privileges": 8,
        ...
    }
}
```

**2.76.1.4.3.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.76.1.4.3.6 Response media types**

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

**2.76.1.4.3.7 Response status**

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.76.1.4.3.8 Response body

 **Note:** The password attribute is not displayed.

 **Example 2-345: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:dm_user xmlns:dm="http://ns.emc.com/documentum"
             xmlns="http://www.w3.org/2005/Atom">
    <dm:attributes>
        <dm:r_object_id>1100000180000101</dm:r_object_id>
        <dm:user_name>acme</dm:user_name>
        <dm:user_os_name>acme</dm:user_os_name>
        <dm:user_address>acme</dm:user_address>
        <dm:user_group_name>docu</dm:user_group_name>
        <dm:user_privileges>16</dm:user_privileges>
        <dm:owner_def_permit>7</dm:owner_def_permit>
        <dm:world_def_permit>8</dm:world_def_permit>
        <dm:group_def_permit>5</dm:group_def_permit>
        <dm:default_folder href="http://localhost:8080/dctm-rest/repositories/acme/users
                           /61636d65/home.xml"/>acme</dm:default_folder>
        <dm:r_is_group>false</dm:r_is_group>
        <dm:user_db_name>acme</dm:user_db_name>
        <dm:acl_domain>acme</dm:acl_domain>
        <dm:acl_name>dm_4500000180000100</dm:acl_name>
        <dm:user_os_domain>2k8vm</dm:user_os_domain>
        <dm:user_state>0</dm:user_state>
        <dm:client_capability>8</dm:client_capability>
        <dm:globally_managed>false</dm:globally_managed>
        <dm:r_modify_date>2012-08-31T22:42:42.000+0800</dm:r_modify_date>
        <dm:workflow_disabled>false</dm:workflow_disabled>
        <dm:alias_set_id>0000000000000000</dm:alias_set_id>
        <dm:user_xprivileges>0</dm:user_xprivileges>
        <dm:r_has_events>false</dm:r_has_events>
        <dm:failed_auth_attempt>0</dm:failed_auth_attempt>
        <dm:user_global_unique_id>2k8vm:acme</dm:user_global_unique_id>
        <dm:user_login_name>acme</dm:user_login_name>
        <dm:user_login_domain>2k8vm</dm:user_login_domain>
        <dm:last_login_utc_time>2012-08-31T14:42:42.000+0800
        </dm:last_login_utc_time>
        <dm:i_is_replica>false</dm:i_is_replica>
        <dm:i_vstamp>0</dm:i_vstamp>
    </dm:attributes>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/acme/users/61636d65.xml"/>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/acme/users/61636d65.xml"/>
    <link rel="delete"
          href="http://localhost:8080/dctm-rest/repositories/acme/users/61636d65.xml"/>
    <link rel="parent"
          href="http://localhost:8080/dctm-rest/repositories/acme/groups.xml?
username=61636d65"/>
    <link rel="http://names.emc.com/link/documentum/relations/default-folder"
          href="http://localhost:8080/dctm-rest/repositories/acme/users/61636d65/
home.xml"/>
</dm:dm_user>
```



 **Example 2-346: JSON response**

```
{
    "attributes" : {
        "r_object_id" : "1100000180000101",
        "user_name" : "acme",
        "user_os_name" : "acme",
```

```
"user_address" : "acme",
"user_group_name" : "docu",
"user_privileges" : 16,
"owner_def_permit" : 7,
"world_def_permit" : 3,
"group_def_permit" : 5,
"default_folder" : "/acme",
"r_is_group" : false,
"user_db_name" : "acme",
"acl_domain" : "acme",
"acl_name" : "dm_4500000180000100",
"user_os_domain" : "2k8vm",
"user_state" : 0,
"client_capability" : 8,
"globally_managed" : false,
"r_modify_date" : "2012-08-31T22:42:42.000+0800",
"workflow_disabled" : false,
"alias_set_id" : "0000000000000000",
"user_xprivileges" : 0,
"r_has_events" : false,
"failed_auth_attempt" : 0,
"user_global_unique_id" : "2k8vm:acme",
"user_login_name" : "acme",
"user_login_domain" : "2k8vm",
"last_login_utc_time" : "2012-08-31T14:42:42.000+0800",
"i_is_replica" : false,
"i_vstamp" : 0
},
"links" : [ {
    "rel" : "self",
    "href" : "http://localhost:8080/dctm-rest/repositories/acme/users/61636d65.json"
}, {
    "rel" : "edit",
    "href" : "http://localhost:8080/dctm-rest/repositories/acme/users/61636d65.json"
}, {
    "rel" : "delete",
    "href" : "http://localhost:8080/dctm-rest/repositories/acme/users/61636d65.json"
}, {
    "rel" : "parent",
    "href" : "http://localhost:8080/dctm-rest/repositories/acme/groups.json?
username=61636d65"
}, {
    "rel" : "http://names.emc.com/link/documentum/relations/default-folder",
    "href" : "http://localhost:8080/dctm-rest/repositories/acme/users/61636d65/home.json"
} ]
}
```



#### 2.76.1.4.4 Delete a User

Only the SysAdmin or SuperUser can perform a delete operation. When a user has been deleted, the deletion may have an impact on the user's authentication.

**2.76.1.4.4.1 HTTP Request method and query parameters**

DELETE

**2.76.1.4.4.2 Request media types**

N/A

**2.76.1.4.4.3 Request headers**

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.76.1.4.4.4 Request body**

N/A

**2.76.1.4.4.5 Response headers**

N/A

**2.76.1.4.4.6 Response media types**

N/A

**2.76.1.4.4.7 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.76.1.4.4.8 Response body**

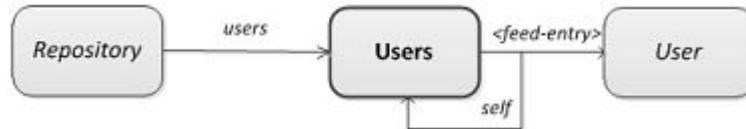
N/A

**2.76.2 Users**

The Users resource represents the metadata of a collection of users in a repository.

### 2.76.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

### 2.76.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: List of users
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “[User](#)” on page 766
- Entry ID: URI of the user
- Entry Title: User name
- Entry Summary: User description
- Entry Updated: The `r_modify_date` of the user

### 2.76.2.3 Link relations

Link relation	Description	Resource reference
self	This collection of Users	<a href="#">“Users” on page 775</a>
first, last, next, previous	Pagination links	<a href="#">“Users” on page 775</a>

### 2.76.2.4 Operations

#### 2.76.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of Users and their metadata
POST	Creates a new User in a repository

#### 2.76.2.4.2 GET Users

Retrieve the metadata of a collection of users with the specified filters.

##### 2.76.2.4.2.1 HTTP Request method and query parameters

GET

This method supports the following common query parameters:

- sort
- view

*Please see the Note below.*
- inline

*Please see the Note below.*
- page
- items-per-page
- include-total
- links
- filter



**Note:** When a collection of Users is retrieved with the parameter `inline=true`, the Users feed will contain the default attribute values for the Users retrieved. One of these attributes, the `definition` property, which is used to identify the entity's type, will always be set to `dm_user` by default. This is the case even when the entity is in fact a subtype of `dm_user`.

To retrieve the correct entity type definition for the Users retrieved, you must set the `view=:all` parameter to retrieve Users.

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.76.2.4.2.2 Request media types

N/A

#### 2.76.2.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.76.2.4.2.4 Request body

N/A

#### 2.76.2.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.76.2.4.2.6 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

#### 2.76.2.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.76.2.4.2.8 Response body

XML or JSON representation of the collection of users.



**Note:** The user name in the URL is encoded.

#### 2.76.2.4.3 Create a user

Create a new user in a repository. The Sysadmin, SuperUser, or any other users that have the <Create Group> privilege can create a user.

##### 2.76.2.4.3.1 HTTP Request method and query parameters

POST

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters on page 1047](#).

##### 2.76.2.4.3.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/xml
- application/json

##### 2.76.2.4.3.3 Request headers

- Accept
- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.76.2.4.3.4 Request body

- The <user\_name> property is mandatory
- The <user\_login\_name> or <user\_os\_name> is mandatory

##### ➡ Example 2-347: Request body in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:dm_user xmlns:dm="http://ns.emc.com/documentum" xmlns="http://www.w3.org/2005/Atom">
    <dm:properties>
        <dm:user_name>acme</dm:user_name>
        <dm:user_login_name>acme</dm:user_login_name>
        // ...
    </dm:properties>
</dm:dm_user>
```



##### ➡ Example 2-348: Request body in JSON

```
{
    "properties" : {
        "user_name" : "acme",
        "user_login_name" : "acme"
```

```
// ...
}
```



#### 2.76.2.4.3.5 Response headers

- Location - The URL of the newly created user.

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.76.2.4.3.6 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.76.2.4.3.7 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.76.2.4.3.8 Response body

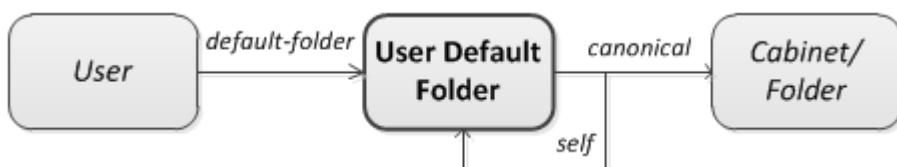
N/A

## 2.77 User Default Folder

The User Default Folder resource represents the default folder (the home cabinet) for a given user.

### 2.77.1 Resource relationships

The following diagram illustrates how this resource is related to other resources.



*See Also:* “User” on page 766 and [About the Diagram on page 13](#).

## 2.77.2 Feed

No feed support.

## 2.77.3 Link relations

Link relation	Description	Resource reference
self	This default folder.	"User Default Folder" on page 780
canonical	Canonical URI for the default folder resource.	"Cabinet" on page 132 "Folder" on page 233

## 2.77.4 Operations

### 2.77.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the default folder for a given user.

### 2.77.4.2 Get a user default folder

Retrieve information about the default folder for a given user. Typically, the default folder is a cabinet, a type of `dm_cabinet`, and is referred to as the "user's home cabinet". The default folder can also be an instance of `dm_folder` or one of its subtypes.

#### 2.77.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameter	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• <code>view</code></li> <li>• <code>permissions</code></li> </ul>	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

#### 2.77.4.2.2 Request media types

- `application/vnd.emc.documentum+xml`
- `application/vnd.emc.documentum+json`

#### 2.77.4.2.3 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.77.4.2.4 Request body

N/A

#### 2.77.4.2.5 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.77.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.77.4.2.7 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.77.4.2.8 Response body

XML or JSON representation of the specified user's default folder:

- The body must contain information about the default folder of the specified user.
- The body may contain all attributes of the default folder, depending on the value that is set for the `view` parameter.

Here is a sample of a typical Request and response.

##### ➡ Example 2-349: XML request

```
Accept: application/vnd.emc.documentum+xml  
GET http://localhost:8080/dctm-rest/repositories/acme01/users/<username>/home
```



▶ **Example 2-350: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<dm:dm_cabinet xmlns:dm="http://ns.emc.com/documentum" xmlns="http://www.w3.org/2005/Atom">
  <dm:attributes>
    <dm:r_object_id
      href="http://localhost:8080/dctm-rest/repositories/acme01/objects/0c00000c80000104.xml">
        0c00000c80000104
    </dm:r_object_id>
    <dm:object_name>acme01</dm:object_name>
    <dm:title>Super User Cabinet</dm:title>
    ...
    <dm:r_object_type>dm_cabinet</dm:r_object_type>
    <dm:r_creation_date>2012-06-13T05:36:08.000+0800</dm:r_creation_date>
    <dm:r_modify_date>2012-06-13T05:36:08.000+0800</dm:r_modify_date>
  </dm:attributes>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/acme01/users/john/home.xml"/>
  <link rel="canonical"
    href="http://localhost:8080/dctm-rest/repositories/acme01/cabinets/0c00000c80000104.xml"/>
    <link rel="http://names.emc.com/link/documentum/relations/folders"
      href="http://localhost:8080/dctm-rest/repositories/acme01/cabinets/0c00000c80000104/folders.xml"/>
    <link rel="http://names.emc.com/link/documentum/relations/documents"
      href="http://localhost:8080/dctm-rest/repositories/acme01/cabinets/0c00000c80000104/documents.xml"/>
    <link rel="http://names.emc.com/link/documentum/relations/objects"
      href="http://localhost:8080/dctm-rest/repositories/acme01/cabinets/0c00000c80000104/objects.xml"/>
  </dm:dm_cabinet>
```



▶ **Example 2-351: JSON request**

```
Accept: application/vnd.emc.documentum+json
GET http://localhost:8080/dctm-rest/repositories/acme01/users/<username>/home
```



▶ **Example 2-352: JSON response**

```
{
  {
    "attributes": {
      "r_object_id": "0c00000c80000104",
      "object_name": "acme01",
      "title": "Super User Cabinet",
      "subject": "",
      ...
    },
    "links": [
      {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/repo/users/Administrator/home.json"
      },
    ]
  }
}
```

```
{
    "rel": "canonical",
    "href": "http://localhost:8080/dctm-rest/repositories/acme01/cabinets/
0c000000c80000104.json"
}
...
}
```



## 2.78 User Permission Set

The User Permission Set resource represents the associated permission set of a user.

Documentum CM Server uses the `default_acl` property of the server configuration object to specify which ACL the server uses as the default ACL. When creating a new object, if an ACL is not explicitly associated with the object you can use the following values:

- <1> The ACL associated with the object's primary folder
- <2> The ACL associated with the object's type
- <3> The ACL associated with the user who created the object
- <4> No default ACL is specified

The default is <3>

When the `default_acl` property of the server configuration object has a value of <3>, the creator's default permissions (or ACL) are used as the default ACL of the new object. This is the case when an ACL is not explicitly associated with the object.

When you have the associated permission set of a user, you can view the default permission set of all the sysobjects that were created by that user. You can also see any change of that user's permission set to determine how that change may effect the default permission set of any sysobjects that are created by that user in the future.

### 2.78.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “User” on page 766 and About the Diagram on page 13.

## 2.78.2 Feed

No feed support.

## 2.78.3 Link relations

The following table lists link relations for the User Permission Set resource.

Link relation	Description	Resource reference
self	This user permission set resource	"User Permission Set" on page 784
acl [1]	The current user's associated ACL object	"ACL" on page 37

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
<http://identifiers.emc.com/linkrel/>.

## 2.78.4 Operations

### 2.78.4.1 Supported HTTP methods

Method	Description
GET	GET the permission set for a specific user

### 2.78.4.2 Get the User Permission Set

Retrieve the definition of the associated permission set for a specific user. The user permission set has the same representation as the object's permission set. The permission set returned may contain one or more sections of the complete permission set, which includes the *<permitted>*, *<restricted>*, *<required-group>*, and *<required-group-set>* sections. Sections that do not appear in the permission set returned are not set by that permission set.

#### 2.78.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Default value
GET	links	For more information, see Appendix B, REST Common Definition - URI Request Query Parameters on page 1047.		

**2.78.4.2.2 Request media types**

N/A

**2.78.4.2.3 Request headers**

- Accept
- Authorization
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.78.4.2.4 Request body**

N/A

**2.78.4.2.5 Response headers**

- Content-Type
- For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.78.4.2.6 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.78.4.2.7 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.78.4.2.8 Response body** **Example 2-353: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<permission-set xmlns="http://identifiers.emc.com/vocab/documentum">
    <permitted>
        <permission accessor="dm_world" basic-permission="Read"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        <permission accessor="dm_owner" basic-permission="Delete"
            extend-permissions="EXECUTE_PROC,CHANGE_LOCATION"/>
        .....
    </permitted>
    <links>
        <link rel="self" href="{userPermissionSetResourceUri}" />
        <link rel="http://identifiers.emc.com/linkrel/acl" />
    </links>
</permission-set>
```

```

        href="{aclResourceUri}"/>
    </links>
</permission-set>

```



#### Example 2-354: JSON response

```

{
    "permitted": [
        {
            "accessor": "dm_world",
            "basic-permission": "Read",
            "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION"
        },
        {
            "accessor": "dm_owner",
            "basic-permission": "Delete",
            "extend-permissions": "EXECUTE_PROC,CHANGE_LOCATION"
        },
        .....
    ],
    "links": [
        {
            "rel": "self",
            "href": "{userPermissionSetResourceUri}"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/acl",
            "href": "{aclResourceUri}"
        }
    ]
}

```



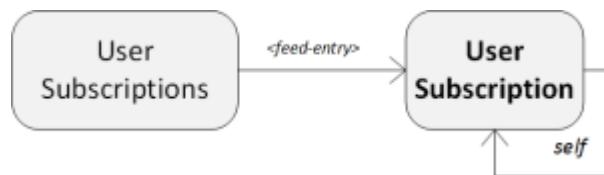
## 2.79 User Subscription (s)

### 2.79.1 User Subscription

The User Subscription resource allows you to subscribe to a sysobject, document, folder, or cabinet. You can use this resource to subscribe to favorite resources.

#### 2.79.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also: “User Subscriptions” on page 798 and About the Diagram on page 13.*

### 2.79.1.2 Feed

No feed support.

### 2.79.1.3 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">"User Subscription" on page 787</a>
relations	The collection of relations related to this object	<a href="#">"Relations" on page 538</a>
unsubscribe	Unsubscribe to this sysobject	<a href="#">"User Subscription" on page 787</a>
parent-links [1]	The collection of parents.   <b>Note:</b> Does not apply to cabinets.	<a href="#">"Parent Links" on page 499</a>
comments [1]	Comments associated to this object	<a href="#">"Comments" on page 156</a>
virtual-document-nodes [1]	GET virtual document nodes when the sysobject is a virtual document	<a href="#">"Virtual Document Nodes" on page 821</a>
virtual-document-conversion [1]	Convert a simple document to virtual one by adding a new node	<a href="#">"Virtual Document Nodes" on page 821</a>
snapshots [1]	GET the snapshots associated with this virtual document   <b>Note:</b> This link only available for virtual documents.	<a href="#">"Virtual Document Snapshots" on page 839</a>
object-aspects [1]	Attached aspect of the sysobject	<a href="#">"Object Aspect" on page 452</a>
permission-set [1]	Permission set of the sysobject	<a href="#">"Permission Set" on page 507</a>
permissions [1]	Specified user's permissions on the given sysobject	<a href="#">"Permissions" on page 503</a>
attachable-lifecycles [1]	Collection of lifecycles to which the given sysobject can be attached	<a href="#">"Lifecycles" on page 385</a>
object-lifecycle [1]	The current lifecycle state of the given sysobject	<a href="#">"Object Lifecycle" on page 460</a>
audit-trails	The audit trails associated with this object	<a href="#">"Audit Trails" on page 102</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		



**Note:** The link relations for this resource are the same as those for the SysObject resource.

## 2.79.1.4 Operations

### 2.79.1.4.1 Supported HTTP methods

Method	Description
PUT	Subscribes users to the object
DELETE	Unsubscribes the current user from the object

### 2.79.1.4.2 Subscribe to an object

#### 2.79.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

#### 2.79.1.4.2.2 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.79.1.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.79.1.4.2.4 Request body

The value of the subscribed user is that user's username or its URI.

#### ➡ Example 2-355: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<subscribers>
    <subscriber>someUser</subscriber>
</subscribers>
```



#### ➡ Example 2-356: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subscribers>
    <subscriber>http://localhost:8080/dctm-rest/repositories/REPO/users/someUser</subscriber>
</subscribers>
```



▶ **Example 2-357: JSON request**

```
{  
    "subscribers": ["someUser"]  
}
```



▶ **Example 2-358: JSON response**

```
{  
    "subscribers": ["http://localhost/dctm-rest/repositories/REPO/users/someUser"]  
}
```



#### 2.79.1.4.2.5 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.79.1.4.2.6 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.79.1.4.2.7 Response status

- 201- Subscribed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.79.1.4.2.8 Response body

▶ Example 2-359: XML

```

<?xml version='1.0' encoding='UTF-8'?>
<object xmlns="http://identifiers.emc.com/vocab/documentum"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="dm_document"
         definition="http://localhost:8080/dctm-rest/repositories/repo/types/dm_document">
<properties>
    <object_name>ff</object_name>
    <r_object_type>dm_document</r_object_type>
    <title></title>
    <subject></subject>
    <authors xsi:nil="true"/>
    <keywords xsi:nil="true"/>
    <a_application_type></a_application_type>
    <a_status></a_status>
    <r_creation_date>2016-12-15T02:33:24.000+00:00</r_creation_date>
    <r_modify_date>2017-09-28T02:53:13.000+00:00</r_modify_date>
    <r_modifier>Administrator</r_modifier>
    <r_access_date xsi:nil="true"/>
    <a_is_hidden>false</a_is_hidden>
    <i_is_deleted>false</i_is_deleted>
    <a_retention_date xsi:nil="true"/>
    <a_archive>false</a_archive>
    <a_compound_architecture></a_compound_architecture>
    <a_link_resolved>false</a_link_resolved>
    <i_reference_cnt>1</i_reference_cnt>
    <i_has_folder>true</i_has_folder>
    <i_folder_id>
        <item>0c02791780009423</item>
    </i_folder_id>
    <r_composite_id xsi:nil="true"/>
    <r_composite_label xsi:nil="true"/>
    <r_component_label xsi:nil="true"/>
    <r_order_no xsi:nil="true"/>
    <r_link_cnt>0</r_link_cnt>
    <r_link_high_cnt>0</r_link_high_cnt>
    <r_assembled_from_id>0000000000000000</r_assembled_from_id>
    <r_frzn_assembly_cnt>0</r_frzn_assembly_cnt>
    <r_has_frzn_assembly>false</r_has_frzn_assembly>
    <resolution_label></resolution_label>
    <r_is_virtual_doc>0</r_is_virtual_doc>
    <i_contents_id>060279178002fab2</i_contents_id>
    <a_content_type>jpeg</a_content_type>
    <r_page_cnt>1</r_page_cnt>
    <r_content_size>18040</r_content_size>
    <a_full_text>true</a_full_text>
    <a_storage_type>filestore_01</a_storage_type>
    <i_cabinet_id>0c02791780009423</i_cabinet_id>
    <owner_name>dmadmin</owner_name>
    <owner_permit>7</owner_permit>
    <group_name></group_name>
    <group_permit>5</group_permit>
    <world_permit>3</world_permit>
    <i_antecedent_id>0000000000000000</i_antecedent_id>
    <i_chronicle_id>09027917800106dd</i_chronicle_id>
    <i_latest_flag>true</i_latest_flag>
    <r_lock_owner></r_lock_owner>
    <r_lock_date xsi:nil="true"/>
    <r_lock_machine></r_lock_machine>
    <log_entry></log_entry>
    <r_version_label>
        <item>1.0</item>
        <item>CURRENT</item>
    </r_version_label>
    <i_branch_cnt>0</i_branch_cnt>
    <i_direct_dsc>false</i_direct_dsc>
    <r_immutable_flag>false</r_immutable_flag>

```

```
<r_frozen_flag>false</r_frozen_flag>
<r_has_events>false</r_has_events>
<acl_domain>dadmin</acl_domain>
<acl_name>dm_4502791780002500</acl_name>
<a_special_app></a_special_app>
<i_is_reference>false</i_is_reference>
<r_creator_name>dadmin</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_effective_date xsi:nil="true"/>
<a_expiration_date xsi:nil="true"/>
<a_publish_formats xsi:nil="true"/>
<a_effective_label xsi:nil="true"/>
<a_effective_flag xsi:nil="true"/>
<a_category></a_category>
<language_code></language_code>
<a_is_template>false</a_is_template>
<a_controlling_app></a_controlling_app>
<r_full_content_size>18040.0</r_full_content_size>
<a_extended_properties xsi:nil="true"/>
<a_is_signed>false</a_is_signed>
<a_last_review_date xsi:nil="true"/>
<i_retain_until xsi:nil="true"/>
<r_aspect_name xsi:nil="true"/>
<i_retainer_id xsi:nil="true"/>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>17</i_vstamp>
<r_object_id>09027917800106dd</r_object_id>
</properties>
<links>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
                      subscription"/>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd"/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd"/>
    <link rel="canonical"
          href="http://localhost:8080/dctm-rest/repositories/repo/documents/
09027917800106dd"/>
    <link rel="http://identifiers.emc.com/linkrel/parent-links"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
                      parent-links"/>
    <link rel="parent" title="0c02791780009423"
          href="http://localhost:8080/dctm-rest/repositories/repo/folders/
0c02791780009423"/>
    <link rel="http://identifiers.emc.com/linkrel/cabinet"
          href="http://localhost:8080/dctm-rest/repositories/repo/cabinets/
0c02791780009423"/>
    <link rel="contents"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
                      contents"/>
    <link rel="http://identifiers.emc.com/linkrel/primary-content"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
                      contents/content"/>
    <link rel="http://identifiers.emc.com/linkrel/checkout"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/lock"/>
    <link rel="version-history"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/
```

```

09027917800106dd/
    versions"/>
<link rel="http://identifiers.emc.com/linkrel/current-version"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    versions/current"/>
<link rel="http://identifiers.emc.com/linkrel/virtual-document-conversion"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    vd-nodes"/>
<link rel="http://identifiers.emc.com/linkrel/relations"
      href="http://localhost:8080/dctm-rest/repositories/repo/relations?
        related-object-id=09027917800106dd&related-object-role=any"/>
<link rel="http://identifiers.emc.com/linkrel/object-aspects"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    aspects"/>
<link rel="http://identifiers.emc.com/linkrel/permission-set"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    permission-set"/>
<link rel="http://identifiers.emc.com/linkrel/permissions"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    permissions"/>
<link rel="http://identifiers.emc.com/linkrel/comments"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    comments"/>
<link rel="http://identifiers.emc.com/linkrel/attachable-lifecycles"
      href="http://localhost:8080/dctm-rest/repositories/repo/lifecycles?
        object-id=09027917800106dd"/>
<link rel="http://identifiers.emc.com/linkrel/object-lifecycle"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    lifecycle"/>
<link rel="http://identifiers.emc.com/linkrel/audit-trails"
      href="http://localhost:8080/dctm-rest/repositories/repo/audit-trails?
        filter=audited_obj_id='09027917800106dd'"/>
<link rel="http://identifiers.emc.com/linkrel/unsubscribe"
      href="http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
    subscription"/>
</links>
</object>

```



### Example 2-360: JSON

```
{
  "name": "object",
  "type": "dm_document",
  "definition": "http://localhost:8080/dctm-rest/repositories/repo/types/dm_document",
  "properties": {
    "object_name": "ff",
    "r_object_type": "dm_document",
    "title": "",
    "subject": "",
    "authors": null,
    "keywords": null,
    "a_application_type": "",
    "a_status": "",
    "r_creation_date": "2016-12-15T02:33:24.000+00:00",
    "r_modify_date": "2017-09-28T02:52:10.000+00:00",
    "r_modifier": "Administrator",
    "r_access_date": null,
    "a_is_hidden": false,
  }
}
```

```
"i_is_deleted": false,
"a_retention_date": null,
"a_archive": false,
"a_compound_architecture": "",
"a_link_resolved": false,
"i_reference_cnt": 1,
"i_has_folder": true,
"i_folder_id": [
    "0c02791780009423"
],
"r_composite_id": null,
"r_composite_label": null,
"r_component_label": null,
"r_order_no": null,
"r_link_cnt": 0,
"r_link_high_cnt": 0,
"r_assembled_from_id": "0000000000000000",
"r_frnz_nAssembly_cnt": 0,
"r_has_frnz_nAssembly": false,
"resolution_label": "",
"r_is_virtual_doc": 0,
"i_contents_id": "060279178002fab2",
"a_content_type": "jpeg",
"r_page_cnt": 1,
"r_content_size": 18040,
"a_full_text": true,
"a_storage_type": "filestore_01",
"i_cabinet_id": "0c02791780009423",
"owner_name": "dmadmin",
"owner_permit": 7,
"group_name": "",
"group_permit": 5,
"world_permit": 3,
"i_antecedent_id": "0000000000000000",
"i_chronicle_id": "09027917800106dd",
"i_latest_flag": true,
"r_lock_owner": "",
"r_lock_date": null,
"r_lock_machine": "",
"log_entry": "",
"r_version_label": [
    "1.0",
    "CURRENT"
],
"i_branch_cnt": 0,
"i_direct_dsc": false,
"r_immutable_flag": false,
"r_frozen_flag": false,
"r_has_events": false,
"acl_domain": "dmadmin",
"acl_name": "dm_4502791780002500",
"a_special_app": "",
"i_is_reference": false,
"r_creator_name": "dmadmin",
"r_is_public": true,
"r_policy_id": "0000000000000000",
"r_resume_state": 0,
"r_current_state": 0,
"r_alias_set_id": "0000000000000000",
"a_effective_date": null,
"a_expiration_date": null,
"a_publish_formats": null,
"a_effective_label": null,
"a_effective_flag": null,
"a_category": "",
"language_code": "",
"a_is_template": false,
"a_controlling_app": "",
"r_full_content_size": 18040,
"a_extended_properties": null,
```

```

    "a_is_signed": false,
    "a_last_review_date": null,
    "i_retain_until": null,
    "r_aspect_name": null,
    "i_retainer_id": null,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 16,
    "r_object_id": "09027917800106dd"
},
"links": [
{
    "rel": "self",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
        subscription"
},
{
    "rel": "edit",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd"
},
{
    "rel": "http://identifiers.emc.com/linkrel/delete",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd"
},
{
    "rel": "canonical",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/documents/
09027917800106dd"
},
{
    "rel": "http://identifiers.emc.com/linkrel/parent-links",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/
        parent-links"
},
{
    "rel": "parent",
    "title": "0c02791780009423",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/folders/
0c02791780009423"
},
{
    "rel": "http://identifiers.emc.com/linkrel/cabinet",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/cabinets/
0c02791780009423"
},
{
    "rel": "contents",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/contents"
},
{
    "rel": "http://identifiers.emc.com/linkrel/primary-content",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/contents/
        content"
},
{
    "rel": "http://identifiers.emc.com/linkrel/checkout",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/lock"
},
{
    "rel": "version-history",
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/
09027917800106dd/versions"
},

```

```
{  
    "rel": "http://identifiers.emc.com/linkrel/current-version",  
    "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/versions/  
        current"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/virtual-document-conversion",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/vd-nodes"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/relations",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/relations?  
            related-object-id=09027917800106dd&related-object-role=any"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/object-aspects",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/aspects"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/permission-set",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/  
            permission-set"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/permissions",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/  
            permissions"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/comments",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/comments"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/attachable-lifecycles",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/lifecycles?  
            object-id=09027917800106dd"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/object-lifecycle",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/lifecycle"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/audit-trails",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/audit-trails?  
            filter=audited_obj_id='09027917800106dd'"  
,  
    {  
        "rel": "http://identifiers.emc.com/linkrel/unsubscribe",  
        "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/  
09027917800106dd/  
            subscription"  
    }  
}  
}
```



**2.79.1.4.3 Unsubscribe from an object****2.79.1.4.3.1 HTTP Request method and query parameters**

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE		For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

**2.79.1.4.3.2 Request media types**

N/A

**2.79.1.4.3.3 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

**2.79.1.4.3.4 Request body**

N/A

**2.79.1.4.3.5 Response headers**

N/A

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

**2.79.1.4.3.6 Response media types**

N/A

**2.79.1.4.3.7 Response status**

- 204- Unsubscribed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

#### 2.79.1.4.3.8 Response body

N/A

### 2.79.2 User Subscriptions

This resource represents a collection of objects to which the current user has subscribed.

#### 2.79.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “[Repository](#)” on page 564 and [About the Diagram](#) on page 13.

#### 2.79.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: /repositories/<repositoryName>/subscriptions
- Feed Title: My Subscriptions
- Feed Updated: Server's current time
- Supports POST: No
- Entry: [SysObject Resource](#)
- Entry ID: The sysobject resource URI
- Entry Title: The id of subscription related object
- Entry Updated: The updated time of the subscription related object

#### 2.79.2.3 Link relations

Link relation	Description	Resource reference
self	This resource	<a href="#">“User Subscriptions” on page 798</a>

## 2.79.2.4 Operations

### 2.79.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a listing of all objects to which the current user has subscribed.

### 2.79.2.4.2 Get all subscriptions

This operation retrieves all of the objects to which the current user has subscribed.

#### 2.79.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• view</li> <li>• inline</li> <li>• items-per-page</li> <li>• page</li> <li>• include-total</li> <li>• filter</li> </ul>	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

#### 2.79.2.4.2.2 Request media types

N/A

#### 2.79.2.4.2.3 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

**2.79.2.4.2.4 Request body**

N/A

**2.79.2.4.2.5 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.79.2.4.2.6 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.79.2.4.2.7 Response status**

- 200 - Retrieved Successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

**2.79.2.4.2.8 Response body****Example 2-361: XML**

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/repo/subscriptions</id>
  <title>My Subscriptions</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2017-09-28T03:24:38.863+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">100</dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/repo/subscriptions"/>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/repo/objects/09027917800106dd</id>
    <title>Subscription Samples</title>
    <author>
      <name>dmadmin</name>
      <uri>http://localhost:8080/dctm-rest/repositories/repo/users/dmadmin</uri>
    </author>
    <summary>dm_document 09027917800106dd</summary>
    <updated>2017-09-28T02:53:13.000+00:00</updated>
    <published>2016-12-15T02:33:24.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/repo/objects/09027917800106dd"/>
    <content type="application/xml"
             src="http://localhost:8080/dctm-rest/repositories/repo/objects/09027917800106dd"/>
```

```
</entry>
</feed>
```



### ➤ Example 2-362: JSON

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/repo/subscriptions",
  "title": "My Subscriptions",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2017-09-28T03:23:16.470+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/repo/subscriptions"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/repo/objects/09027917800106dd",
      "title": "Subscription Samples",
      "author": [
        {
          "name": "dmadmin",
          "uri": "http://localhost:8080/dctm-rest/repositories/repo/users/dmadmin"
        }
      ],
      "summary": "dm_document 09027917800106dd",
      "updated": "2017-09-28T02:53:13.000+00:00",
      "published": "2016-12-15T02:33:24.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/repo/objects/09027917800106dd"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/repo/objects/09027917800106dd"
      }
    }
  ]
}
```

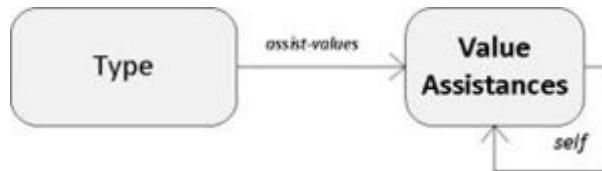


## 2.80 Value Assistsances

The Value Assistsances resource represents the value assistances of a Type.

### 2.80.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



*See Also:* “Type” on page 727 and About the Diagram on page 13.

### 2.80.2 Feed

No feed support.

### 2.80.3 Link relations

Link relation	Description	Resource reference
self	This Value Assistsances resource	“Value Assistsances” on page 802

### 2.80.4 Operations

#### 2.80.4.1 Supported HTTP methods

Method	Description
POST	Retrieves the Value Assistance and dependency properties of the specified Type

#### 2.80.4.2 Get the value assistance of the type

GET the Value Assistance and dependency properties of the specified type

#### 2.80.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	included-properties	The property names to be included in the Response.  The property names are comma separated. For example, included-properties =attr1, attr2, attr3	string	N/A	All properties belonging to the Type
	lifecycle-id	The lifecycle id property.	string	N/A	null
	state	The Lifecycle state string.	string	N/A	null

#### 2.80.4.2.2 Request headers

- Authorization
- Accept
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.80.4.2.3 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.80.4.2.4 Request body

The input available for one property of a certain type may depend on the value of other properties. When you want to get the assist values for a property, you must provide the values of dependent properties if there are dependencies.

Resources return multi-properties assist values, therefore you must provide all of the dependent values that the assist values require.

Every property's dependency information is displayed within the Type resource. This information is within the `<dependencies>` element, which is within the `<property>` element.

Here are two code samples (one in XML and one in JSON) that illustrates this point:

##### ➡ Example 2-363: XML request

```
<assist-value-request>
    <!-- Includes all of the dependency values for input -->
    <properties>
        <!-- Some assist values may be dependent upon the value of attr1,
            so this element provides the value of attr1.-->
        <attr1>value1</attr1>

        <!-- When the dependency value is a list, inputting it as an array
            is supported, as shown here in the attr2 element-->
        <attr2>
            <item>XXX</item>
            <item>YYY</item>
        </attr2>
        <attr3>4</attr3>
    </properties>
</assist-value-request>
```



##### ➡ Example 2-364: JSON request

```
{
    // Includes all of the dependency values for input.
    "properties": {
        // Some assist values may be dependent upon the value of attr1,
        // so this element provides the value of attr1.
        "attr1": "value1",

        // When the dependency value is a list, inputting it as an array
        // is supported, as shown here in the attr2 element.
        "attr2": [
            "XXX",
            "YYY"
        ],
        "attr3": 3
    }
}
```



#### 2.80.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.80.4.2.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.80.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.80.4.2.8 Response body

##### ➡ Example 2-365: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<assist-values>
  <properties>
    <attr1 allow-user-values="false">
      <value label="One">1</value>
      <value label="Two">2</value>
    </attr1>
    <attr2 allow-user-values="false">
      <value label="Three">3</value>
      <value label="Four">4</value>
    </attr2>
  </properties>
  <links>
    <link rel="self"
          href="http://localhost:8080/dctm-rest/repositories/REPO/types/a_new_type/
          assist-values.xml"/>
  </links>
</assist-values>
```



##### ➡ Example 2-366: JSON response

```
{
  "properties": {
    "attr1": {
      "allow-user-values": false,
      "values": [
        {"value": 1, "label": "One"},
        {"value": 2, "label": "Two"},
      ]
    },
    "attr2": {
      "allow-user-values": true,
      "values": [
        {"value": 3, "label": "Three"},
```

```

        {"value": 4, "label": "Four"},  

    ]  

},  

"links": [  

{
    "rel": "self",
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/types/a_new_type/
        assist-values.xml"
}
]
}

```



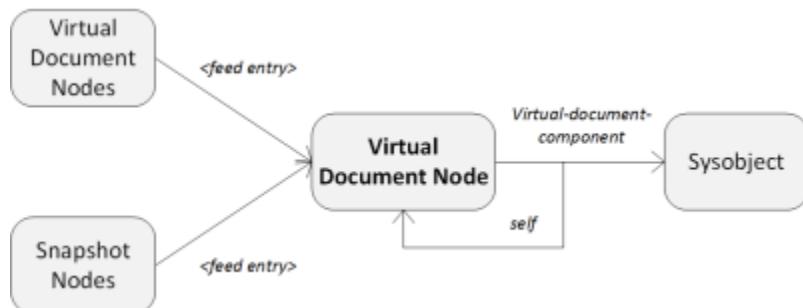
## 2.81 Virtual Document Node (s)

### 2.81.1 Virtual Document Node

This resource can represent a Virtual Document Node or a Snapshot Node in a repository. You can use this resource to retrieve, update, or remove a Virtual Document Node from Documentum CM Server.

#### 2.81.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Virtual Document Nodes” on page 821 and [About the Diagrams on page 13](#).

#### 2.81.1.2 Feed

No feed support.

### 2.81.1.3 Link relations

Link relation	Description	Resource reference
self	This Virtual Document Node resource	"Virtual Document Node" on page 806
edit	Update the Virtual Document Node resource	"Virtual Document Node" on page 806
delete [1]	Delete the Virtual Document Node resource	"Virtual Document Node" on page 806
virtual-document-component [1]	The component object of the Virtual Document Node	"SysObject" on page 692
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

### 2.81.1.4 Operations

#### 2.81.1.4.1 Supported HTTP methods

Method	Description
GET	GET a Virtual Document Node
POST	Update a Virtual Document Node
DELETE	Delete a Virtual Document Node

#### 2.81.1.4.2 Get a virtual document node

Retrieves a Virtual Document Node, its metadata including the parent-id, chronicle-id, relation-id, binding, copy-behavior, follow-assembly, order-number, and the override-late-binding attributes. It also retrieves the properties of the corresponding component.

##### 2.81.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

HTTP method	Query parameters	Description	Data type	Value range	Default value
	object-id	<p>The object id of the virtual document component of the current virtual document node.</p> <p>When the <i>&lt;object-id&gt;</i> parameter is set, the properties of the virtual document component, defined by the <i>&lt;object-id&gt;</i>, is returned.</p> <p>When the <i>&lt;object-id&gt;</i> parameter is not specified, the properties of the CURRENT version virtual document component are returned.</p>	string	N/A	N/A

#### 2.81.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.81.1.4.2.3 Request media types**

N/A

**2.81.1.4.2.4 Request body**

N/A

**2.81.1.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.81.1.4.2.6 Response headers**

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.81.1.4.2.7 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.81.1.4.2.8 Response body**

The following table lists the attributes in the response payload of the virtual document node:

<b>Attribute</b>	<b>Description</b>
<i>parent-id</i>	The <code>r_object_id</code> attribute of the node's parent object.
<i>chronicle-id</i>	The <code>chronicle_id</code> attribute of the node object.
<i>relation-id</i>	The <code>r_object_id</code> attribute of the <code>dmrContainment</code> instance that connects the parent and node.

Attribute	Description
<i>order-number</i>	<p>The <code>order_no</code> attribute of the <code>dmrContainment</code> instance that connects the parent and node.</p> <p> <b>Note:</b> The underlying attribute has a float data type, however REST uses an integer to represent it.</p>
<i>binding</i>	<p>The <code>version_label</code> attribute of the <code>dmrContainment</code> instance that connects the parent and node. It defines the version label of the Virtual Document Node. When this attribute has no value, it indicates that late binding has been defined for the node.</p>
<i>override-late-binding</i>	<p>Returns the <code>use_node_ver_label</code> attribute value from the <code>dmrContainment</code> instance that connects the parent and node. It defines a strategy to resolve the version label of its late binding descendants. This attribute value is important when the node is early-bound to the virtual document and is itself a virtual document. In these cases, the <code>use_node_ver_label</code> attribute value determines whether the system uses the node's early-bound version label to select the node's late-bound children.</p> <p>Returns <code>true</code> when the system can use the node's early-bound version label to select its late-bound children, and <code>false</code> otherwise. The return value of this attribute is always <code>false</code> for assembly nodes and the root node.</p>
<i>copy-behavior</i>	<p>The <code>copy_child</code> attribute of the <code>dmrContainment</code> instance that connects the parent and node. It defines the copy strategy. When you are working with REST services, use UNSPECIFIED, REFERENCE, and DUPLICATION as its values.</p>
<i>follow-assembly</i>	<p>The <code>follow_assembly</code> attribute of the <code>dmrContainment</code> instance that connects the parent and node. It determines whether the server selects component descendants using the containment objects, or a snapshot associated with the component instead. Simply put, this attribute determines whether the server treats the <code>SysObject</code> as a Virtual Document or a Snapshot for the purposes of retrieving its nodes or snapshot nodes accordingly.</p>

▶ **Example 2-367: XML response**

```

<?xml version='1.0' encoding='UTF-8'?>
<vd-node xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="dm_document">
  <parent-id>0900000580004993</parent-id>
  <chronicle-id>0900000580004994</chronicle-id>
  <relation-id>050000058000010c</relation-id>
  <order-number>3000</order-number>
  <binding></binding>
  <override-late-binding>false</override-late-binding>
  <copy-behavior>UNSPECIFIED</copy-behavior>
  <follow-assembly>false</follow-assembly>
  <properties>
    <object_name>restDoc70c04679-bfc6-4ea0-9548-c07184b74742</object_name>
    <r_object_type>dm_document</r_object_type>
    <r_creation_date>2017-07-05T02:30:51.000+00:00</r_creation_date>
    <r_modify_date>2017-07-05T02:30:51.000+00:00</r_modify_date>
    <r_modifier>dave</r_modifier>
    <a_is_hidden>false</a_is_hidden>
    <i_is_deleted>false</i_is_deleted>
    <a_archive>false</a_archive>
    <a_link_resolved>false</a_link_resolved>
    <i_reference_cnt>1</i_reference_cnt>
    <i_has_folder>true</i_has_folder>
    <i_folder_id>
      <item>0c00000580000107</item>
    </i_folder_id>
    <r_link_cnt>0</r_link_cnt>
    <r_link_high_cnt>0</r_link_high_cnt>
    <r_assembled_from_id>0000000000000000</r_assembled_from_id>
    <r_frnz_assembly_cnt>1</r_frnz_assembly_cnt>
    <r_has_frnz_assembly>false</r_has_frnz_assembly>
    <r_is_virtual_doc>0</r_is_virtual_doc>
    <i_contents_id>0000000000000000</i_contents_id>
    <r_page_cnt>0</r_page_cnt>
    <r_content_size>0</r_content_size>
    <a_full_text>true</a_full_text>
    <i_cabinet_id>0c00000580000107</i_cabinet_id>
    <owner_name>dave</owner_name>
    <owner_permit>7</owner_permit>
    <group_permit>5</group_permit>
    <world_permit>3</world_permit>
    <i_antecedent_id>0000000000000000</i_antecedent_id>
    <i_chronicle_id>0900000580004994</i_chronicle_id>
    <i_latest_flag>true</i_latest_flag>
    <r_lock_owner>dmadmin</r_lock_owner>
    <r_lock_date>2017-07-06T03:23:31.000+00:00</r_lock_date>
    <r_lock_machine>cnenwangc31m2.corp.emc.com</r_lock_machine>
    <r_version_label>
      <item>1.0</item>
      <item>CURRENT</item>
    </r_version_label>
    <i_branch_cnt>0</i_branch_cnt>
    <i_direct_dsc>false</i_direct_dsc>
    <r_immutable_flag>true</r_immutable_flag>
    <r_frozen_flag>false</r_frozen_flag>
    <r_has_events>false</r_has_events>
    <acl_domain>dave</acl_domain>
    <acl_name>dm_450000058000d07</acl_name>
    <i_is_reference>false</i_is_reference>
    <r_creator_name>dave</r_creator_name>
    <r_is_public>true</r_is_public>
    <r_policy_id>0000000000000000</r_policy_id>
    <r_resume_state>0</r_resume_state>
    <r_current_state>0</r_current_state>
    <r_alias_set_id>0000000000000000</r_alias_set_id>
    <a_is_template>false</a_is_template>
  </properties>
</vd-node>

```

```

<r_full_content_size>0.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition></i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>1</i_vstamp>
<r_object_id>0900000580004994</r_object_id>
</properties>
<links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
        050000058000010c?object-id=0900000580004994&inline=true"/>
    <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000580004994"/>
    <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
        050000058000010c?object-id="/>
    <link rel="http://identifiers.emc.com/linkrel/delete"
        href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
        050000058000010c?object-id="/>
</links>
</vd-node>
```



### Example 2-368: JSON response

```
{
    "name": "vd-node",
    "type": "dm_document",
    "parent-id": "0900000580004993",
    "chronicle-id": "0900000580004994",
    "relation-id": "050000058000010c",
    "order-number": 3000,
    "binding": "",
    "override-late-binding": false,
    "copy-behavior": "UNSPECIFIED",
    "follow-assembly": false,
    "properties": {
        "object_name": "restDoc70c04679-bfc6-4ea0-9548-c07184b74742",
        "r_object_type": "dm_document",
        "r_creation_date": "2017-07-05T02:30:51.000+00:00",
        "r_modify_date": "2017-07-05T02:30:51.000+00:00",
        "r_modifier": "dave",
        "a_is_hidden": false,
        "i_is_deleted": false,
        "a_archive": false,
        "a_link_resolved": false,
        "i_reference_cnt": 1,
        "i_has_folder": true,
        "i_folder_id": [
            "0c00000580000107"
        ],
        "r_link_cnt": 0,
        "r_link_high_cnt": 0,
        "r_assembled_from_id": "0000000000000000",
        "r_frnz_n_assembly_cnt": 1,
        "r_has_frnz_n_assembly": false,
        "r_is_virtual_doc": 0,
        "i_contents_id": "0000000000000000",
        "r_page_cnt": 0,
        "r_content_size": 0,
        "a_full_text": true,
        "i_cabinet_id": "0c0000058000107",
        "owner_name": "dave",
        "owner_permit": 7,
        "group_permit": 5
    }
}
```

```

    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "0900000580004994",
    "i_latest_flag": true,
    "r_lock_owner": "dmadmin",
    "r_lock_date": "2017-07-06T03:23:31.000+00:00",
    "r_lock_machine": "cnenwangc31m2.corp.emc.com",
    "r_version_label": [
        "1.0",
        "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": true,
    "r_frozen_flag": false,
    "r_has_events": false,
    "acl_domain": "dave",
    "acl_name": "dm_4500000580000d07",
    "i_is_reference": false,
    "r_creator_name": "dave",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 1,
    "r_object_id": "0900000580004994"
},
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010c?object-id=0900000580004994&inline=true"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004994"
    },
    {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010c?object-id="
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010c?object-id="
    }
]
}

```



#### 2.81.1.4.3 Update a virtual document node

This operation is used to modify the attributes of a virtual document node. Modifying the attributes of virtual document nodes has an impact on the retrieval of those virtual document nodes.

The following attributes can be updated using this operation:

- `order-number`
- `binding`
- `override-late-binding`
- `copy-behavior`
- `follow-assembly`

When you are updating a virtual document node, that node remains within its specific parent virtual document. When attributes that are not to be updated are included as part of an update operation, you must make sure that their defined values align with their existing values. Otherwise, unwanted changes may occur.



##### Caution

Updating the `relation-id`, `parent-id`, or the `chronicle-id` attributes is not supported by Foundation REST API.

You can use the Foundation Java API to update the `parent-id`, and the `chronicle-id` attributes. If you are updating any of these attributes, please note the following items:

- Updating the `parent-id` of a node causes that node to move to another virtual document.
- Updating the `chronicle-id` attribute of a node causes that node to be removed and replaced with another `SysObject`.
- The `relation-id` cannot be updated by Foundation Java API or Foundation REST API.

#### 2.81.1.4.3.1 HTTP Request method and query parameters

POST

#### 2.81.1.4.3.2 Request headers

- Authorization
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.81.1.4.3.3 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.81.1.4.3.4 Request body

A request is the representation of the virtual document node, with the exception of the `relation-id`, the `parent-id`, and the `chronicle-id` attributes.

##### ➡ Example 2-369: XML request

*Request media type: application/vnd.emc.documentum+xml*

```
POST http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010c
Accept: application/vnd.emc.documentum+xml

<dm:vd-node xmlns:dm="http://identifiers.emc.com/vocab/documentum">
  <dm:order-number>3000</dm:order-number>
  <dm:binding>CURRENT</dm:binding>
  <dm:override-late-binding>true</dm:override-late-binding>
  <dm:copy-behavior>UNSPECIFIED</dm:copy-behavior>
  <dm:follow-assembly>false</dm:follow-assembly>
</dm:vd-node>
```



##### ➡ Example 2-370: JSON request

*Request media type: application/vnd.emc.documentum+json*

```
POST http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010c
Accept: application/vnd.emc.documentum+json
{
  "order-number": 3000,
  "binding": "CURRENT",
  "override-late-binding": true,
  "copy-behavior": "UNSPECIFIED",
  "follow-assembly": false
}
```



#### 2.81.1.4.3.5 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.81.1.4.3.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.81.1.4.3.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.81.1.4.3.8 Response body

##### Example 2-371: XML response

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<?xml version='1.0' encoding='UTF-8'?>
<vd-node xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <parent-id>0900000580005555</parent-id>
    <chronicle-id>0900000580005556</chronicle-id>
    <relation-id>0500000580000511</relation-id>
    <order-number>3000</order-number>
    <binding>CURRENT</binding>
    <override-late-binding>true</override-late-binding>
    <copy-behavior>UNSPECIFIED</copy-behavior>
    <follow-assembly>false</follow-assembly>
    <properties>
        <object_name>restDoc70c04679-bfc6-4ea0-9548-c07184b74742</object_name>
        <r_object_type>dm_document</r_object_type>
        <r_creation_date>2017-07-05T02:30:51.000+00:00</r_creation_date>
        <r_modify_date>2017-07-05T02:30:51.000+00:00</r_modify_date>
        <r_modifier>dave</r_modifier>
        <a_is_hidden>false</a_is_hidden>
        <i_is_deleted>false</i_is_deleted>
        <a_archive>false</a_archive>
        <a_link_resolved>false</a_link_resolved>
        <i_reference_cnt>1</i_reference_cnt>
        <i_has_folder>true</i_has_folder>
        <i_folder_id>
            <item>0c00000580000107</item>
        </i_folder_id>
        <r_link_cnt>0</r_link_cnt>
        <r_link_high_cnt>0</r_link_high_cnt>
        <r_assembled_from_id>0000000000000000</r_assembled_from_id>
        <r_frzn_assembly_cnt>1</r_frzn_assembly_cnt>
    </properties>
</vd-node>
```

```

<r_has_frzn_assembly>false</r_has_frzn_assembly>
<r_is_virtual_doc>0</r_is_virtual_doc>
<i_contents_id>0000000000000000</i_contents_id>
<r_page_cnt>0</r_page_cnt>
<r_content_size>0</r_content_size>
<a_full_text>true</a_full_text>
<i_cabinet_id>0c0000580000107</i_cabinet_id>
<owner_name>dave</owner_name>
<owner_permit>7</owner_permit>
<group_permit>5</group_permit>
<world_permit>3</world_permit>
<i_antecedent_id>0000000000000000</i_antecedent_id>
<i_chronicle_id>090000580005556</i_chronicle_id>
<i_latest_flag>true</i_latest_flag>
<r_lock_owner>dmadmin</r_lock_owner>
<r_lock_date>2017-07-06T03:23:31.000+00:00</r_lock_date>
<r_lock_machine>cnenwangc31m2.corp.emc.com</r_lock_machine>
<r_version_label>
  <item>1.0</item>
  <item>CURRENT</item>
</r_version_label>
<i_branch_cnt>0</i_branch_cnt>
<i_direct_dsc>false</i_direct_dsc>
<r_immutable_flag>true</r_immutable_flag>
<r_frozen_flag>false</r_frozen_flag>
<r_has_events>false</r_has_events>
<acl_domain>dave</acl_domain>
<acl_name>dm_4500000580000d07</acl_name>
<i_is_reference>false</i_is_reference>
<r_creator_name>dave</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_is_template>false</a_is_template>
<r_full_content_size>0.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>1</i_vstamp>
<r_object_id>0900000580004994</r_object_id>
</properties>
<links>
  <link rel="self"
    href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
      0500000580000511"/>
  <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
    href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
      0900000580005556"/>
  <link rel="edit"
    href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
      0500000580000511"/>
  <link rel="http://identifiers.emc.com/linkrel/delete"
    href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
      0500000580000511"/>
</links>
</vd-node>

```



### Example 2-372: JSON response

```

Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "name": "vd-node",

```

```
"parent_id": "09000005800048b3",
"chronicle_id": "0900000580002516",
"relation_id": "05000005800010c",
"order_number": 3000,
"binding": "CURRENT",
"override-late-binding": true,
"copy-behavior": "UNSPECIFIED",
"follow-assembly": false,
"properties": {
    "object_name": "restDoc70c04679-bfc6-4ea0-9548-c07184b74742",
    "r_object_type": "dm_document",
    "r_creation_date": "2017-07-05T02:30:51.000+00:00",
    "r_modify_date": "2017-07-05T02:30:51.000+00:00",
    "r_modifier": "dave",
    "a_is_hidden": false,
    "i_is_deleted": false,
    "a_archive": false,
    "a_link_resolved": false,
    "i_reference_cnt": 1,
    "i_has_folder": true,
    "i_folder_id": [
        "0c00000580000107"
    ],
    "r_link_cnt": 0,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "0000000000000000",
    "r_frzn_assembly_cnt": 1,
    "r_has_frzn_assembly": false,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0000000000000000",
    "r_page_cnt": 0,
    "r_content_size": 0,
    "a_full_text": true,
    "i_cabinet_id": "0c00000580000107",
    "owner_name": "dave",
    "owner_permit": 7,
    "group_permit": 5,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "0900000580005556",
    "i_latest_flag": true,
    "r_lock_owner": "dmadmin",
    "r_lock_date": "2017-07-06T03:23:31.000+00:00",
    "r_lock_machine": "cnenwangc31m2.corp.emc.com",
    "r_version_label": [
        "1.0",
        "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": true,
    "r_frozen_flag": false,
    "r_has_events": false,
    "acl_domain": "dave",
    "acl_name": "dm_450000058000d07",
    "i_is_reference": false,
    "r_creator_name": "dave",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 1,
    "r_object_id": "0900000580004994"
},
```

```

    "links": [
      {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          050000058000010c"
      },
      {
        "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
          0900000580002516"
      },
      {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          050000058000010c"
      },
      {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          050000058000010c"
      }
    ]
  }

```



#### 2.81.1.4.4 Delete a virtual document node

This operation removes the current node from the virtual document that is specified in the parent-id attribute.

 **Tip:** When a virtual document has only its root node, deleting that remaining root node converts the virtual document into a simple document.

##### 2.81.1.4.4.1 HTTP Request method and query parameters

DELETE

##### 2.81.1.4.4.2 Request headers

Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.81.1.4.4.3 Request media types

N/A

**2.81.1.4.4.4 Request body**

N/A

**2.81.1.4.4.5 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.81.1.4.4.6 Response headers**

N/A

**2.81.1.4.4.7 Response media types**

N/A

**2.81.1.4.4.8 Response body**

N/A

**2.81.1.4.4.9 Examples****➡ Example 2-373: Delete a child node**

In the following sample we use the ID of a child node (050000058000010c) to define the child node that will be removed.

*Request*

```
DELETE http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010c
```

*Response*

```
Status Code: 204 No Content
```

**➡ Example 2-374: Delete the root node**

In the following sample we use the ID of the root node (\_root\_node\_id\_<VIRTUAL\_DOCUMENT\_ID>) to indicate that the root node will be deleted. When a virtual document has only its root node, deleting that remaining root node converts the virtual document into a simple document.

*Request*

```
DELETE http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/_root_node_id_050000058000010c
```

*Response*



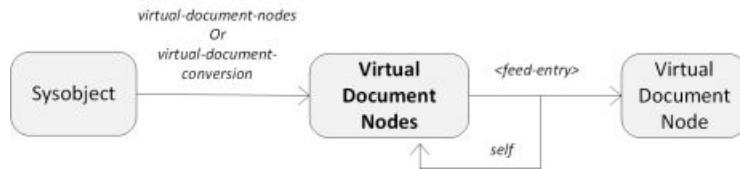
## 2.81.2 Virtual Document Nodes

The Virtual Document Nodes resource represents the virtual document hierarchy of a flattened feed. You can use this resource to [GET the Virtual Document-Hierarchy](#), or [Add a Child Node to a Virtual Document](#).

For information on how to [GET a Virtual Document Node](#), [Update a Virtual Document Node](#), or [Delete a Virtual Document Node](#), see “Virtual Document Node” on page 806.

### 2.81.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[SysObject](#)” on page 692 and [About the Diagrams](#) on page 13.

### 2.81.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: Feed URI
- Feed Title: Virtual Document Nodes
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: {Link to entry resource definition}
- Entry ID: <EntryURI>
- Entry Title: Object name
- Entry Updated: The modified date

### 2.81.2.3 Link relations

Link relation	Description	Resource reference
self	This Virtual Document Nodes resource	<a href="#">"Virtual Document Nodes" on page 821</a>
first, last, next, and previous	Pagination links to additional Virtual Document nodes	<a href="#">"Virtual Document Nodes" on page 821</a>

### 2.81.2.4 Operations

#### 2.81.2.4.1 Supported HTTP methods

Method	Description
GET	Gets the virtual document hierarchy
POST	Add a child node to a virtual document

#### 2.81.2.4.2 Get the virtual document hierarchy

Retrieves the virtual document hierarchy of a specified object. The object can be a virtual document or an assembly object, which is also known as a ["Virtual Document Snapshot" on page 833](#).

##### 2.81.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	binding-label	The symbolic version label to use when resolving late binding nodes.	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	follow-assembly	<p>Determines whether the server treats the SysObject as a Virtual Document or a Snapshot for the purposes of retrieving its nodes or snapshot nodes accordingly.</p> <p>When this parameter is set to <i>true</i>, the server treats the SysObject as a Virtual Document and retrieves its nodes.</p> <p>When this parameter is set to <i>false</i>, the server treats the SysObject as a Snapshot, when a snapshot is available, the server retrieves its snapshot nodes.</p>	string	<i>true, false</i>	<i>false</i>
	include-broken	Determines whether to display nodes with broken bindings using the CURRENT label. When nodes cannot be bound to a Virtual Document with a binding condition, and this parameter is set to <i>true</i> , the nodes are given the CURRENT label and are then bound to the Virtual Document.	string	<i>true, false</i>	<i>false</i>
	depth	Determines the level of depth that the virtual document hierarchy will be returned. Setting a value of -1 will return the whole virtual directory tree.	integer	-1 or >= 0	1

HTTP method	Query parameters	Description	Data type	Value range	Default value
	inline	Whether to include not only the attributes of virtual document node but also the properties of virtual document component in the entry of atom.	string	<p><i>true, false</i></p> <p><i>&lt;true&gt;</i> - Returns the virtual document node and component within the entry's content element</p> <p><i>&lt;false&gt;</i> - Does not return the virtual document node.</p>	<i>false</i>
	page links items-per-page	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			
	vdm-number	The node's number in the virtual document tree. For example 1.2.3	string	N/A	null

#### 2.81.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.81.2.4.2.3 Request media types

N/A

#### 2.81.2.4.2.4 Request body

N/A

#### 2.81.2.4.2.5 Response status

- 200 - Operation was successful
- 400, 401, 403, 404, 500, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.81.2.4.2.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.81.2.4.2.7 Response media types

- application/atom+xml
- application/vnd.emc.documentum+json

#### 2.81.2.4.2.8 Response body

XML or JSON representation of the virtual document.



**Note:** A virtual document, even a childless virtual document, always has at least one node, the root node.

- The response body contains a list of nodes that are present in the virtual document.
- The vdm-number attribute is displayed in the summary property of each node, with the exception of the root node.
- The collection of nodes displayed includes only those nodes to which the REST client has access.
- Pagination is supported.

#### ➤ Example 2-375: XML - GET the virtual document hierarchy

*XML request*

```
GET http://localhost:8080/dctm-rest/repositories/REPO/objects/09000005800048b3/
  vd-nodes?inline=true
Accept: application/atom+xml
```

*XML response*

```

Status Code: 200 OK
Content-Type: application/atom+xml; charset=UTF-8

<?xml version='1.0' encoding='utf-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>
    http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004959/vd-nodes</id>
  <title>Virtual Document Nodes</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2017-04-24T02:14:56.531+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
  100</dm:items-per-page>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000580004959/vd-nodes?inline=true" />
  <entry>
    <id>0900000580004959</id>
    <title>node-0</title>
    <updated>2017-04-24T02:11:51.000+00:00</updated>
    <published>2017-04-24T02:11:49.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          _root_node_id_0900000580004959" />
    <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
          href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
          0900000580004959" />
    <content>
      <dm:vd-node xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        <dm:selected-object-name>node-0</dm:selected-object-name>
        <dm:selected-object-id>0900000580004959</dm:selected-object-id>
        <dm:parent-id>0900000580004959</dm:parent-id>
        <dm:chronicle-id>0900000580004959</dm:chronicle-id>
        <dm:relation-id>_root_node_id_0900000580004959</dm:relation-id>
        <dm:order-number>0</dm:order-number>
        <dm:binding>CURRENT</dm:binding>
        <dm:override-late-binding>true</dm:override-late-binding>
        <dm:copy-behavior></dm:copy-behavior>
        <dm:follow-assembly>false</dm:follow-assembly>
        <dm:node-id>0000000000000000</dm:node-id>
        <dm:is-virtual-document>true</dm:is-virtual-document>
        <dm:late-binding-value>CURRENT</dm:late-binding-value>
        <dm:is-binding-broken>false</dm:is-binding-broken>
        <dm:are-children-compound>false</dm:are-children-compound>
        <dm:available-versions>
          <dm:item>CURRENT</dm:item>
          <dm:item>1.0</dm:item>
        </dm:available-versions>
        <dm:can-be-removed>true</dm:can-be-removed>
        <dm:can-be-restructured>true</dm:can-be-restructured>
        <dm:child-count>1</dm:child-count>
        <dm:is-compound>false</dm:is-compound>
        <dm:is-from-assembly>false</dm:is-from-assembly>
        <dm:is-structurally-modified>false</dm:is-structurally-modified>
        <dm:links>
          <dm:link rel="self"
                  href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                  _root_node_id_0900000580004959" />
          <dm:link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
                  href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
                  0900000580004959" />
          <dm:link rel="edit"
                  href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                  _root_node_id_0900000580004959" />
        </dm:links>
      </dm:vd-node>
    </content>
  </entry>
</feed>

```

```

        <dm:link rel="http://identifiers.emc.com/linkrel/delete"
            href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
            _root_node_id_0900000580004959" />
    </dm:links>
    </dm:vd-node>
</content>
</entry>
<entry>
    <id>0900000580004964</id>
    <title>node-1.1</title>
    <summary>1</summary>
    <updated>2017-04-24T02:11:50.000+00:00</updated>
    <published>2017-04-24T02:11:50.000+00:00</published>
    <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
        050000058000010b" />
    <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
        href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        0900000580004964" />
<content>
    <dm:vd-node xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        <dm:selected-object-name>node-1.1.</dm:selected-object-name>
        <dm:selected-object-id>0900000580004964</dm:selected-object-id>
        <dm:parent-id>0900000580004959</dm:parent-id>
        <dm:chronicle-id>090000058000495a</dm:chronicle-id>
        <dm:relation-id>050000058000010b</dm:relation-id>
        <dm:order-number>1000</dm:order-number>
        <dm:binding>2.0.</dm:binding>
        <dm:override-late-binding>true</dm:override-late-binding>
        <dm:copy-behavior>0</dm:copy-behavior>
        <dm:follow-assembly>false</dm:follow-assembly>
        <dm:node-id>0000000000000001</dm:node-id>
        <dm:vd-number>1</dm:vd-number>
        <dm:is-virtual-document>false</dm:is-virtual-document>
        <dm:late-binding-value>2.0.</dm:late-binding-value>
        <dm:is-binding-broken>false</dm:is-binding-broken>
        <dm:are-children-compound>false</dm:are-children-compound>
        <dm:available-versions>
            <dm:item>CURRENT</dm:item>
            <dm:item>3.0.</dm:item>
            <dm:item>2.0.</dm:item>
            <dm:item>1.0.</dm:item>
        </dm:available-versions>
        <dm:can-be-removed>true</dm:can-be-removed>
        <dm:can-be-restructured>true</dm:can-be-restructured>
        <dm:child-count>0</dm:child-count>
        <dm:is-compound>false</dm:is-compound>
        <dm:is-from-assembly>false</dm:is-from-assembly>
        <dm:is-structurally-modified>false</dm:is-structurally-modified>
        <dm:links>
            <dm:link rel="self"
                href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                050000058000010b" />
            <dm:link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
                href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
                090000058000495a" />
            <dm:link rel="edit"
                href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                050000058000010b" />
            <dm:link rel="http://identifiers.emc.com/linkrel/delete"
                href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                050000058000010b" />
        </dm:links>
    </dm:vd-node>
</content>
</entry>
</feed>
```



 **Example 2-376: JSON - GET the virtual document hierarchy**

*JSON request*

```
GET http://localhost:8080/dctm-rest/repositories/REPO/objects/09000005800048b3/
    vd-nodes?inline=true
Accept: application/vnd.emc.documentum+json
```

*JSON response*

```
Status Code: 200 OK
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004959/
    vd-nodes",
  "title": "Virtual Document Nodes",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2017-04-24T02:15:26.498+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580004959/
        vd-nodes?inline=true"
    }
  ],
  "entries": [
    {
      "id": "0900000580004959",
      "title": "node-0",
      "updated": "2017-04-24T02:11:51.000+00:00",
      "published": "2017-04-24T02:11:49.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
            _root_node_id_0900000580004959"
        },
        {
          "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000580004959"
        }
      ],
      "content": {
        "name": "vd-node",
        "selected-object-name": "node-0",
        "selected-object-id": "0900000580004959",
        "parent-id": "0900000580004959",
        "chronicle-id": "0900000580004959",
        "relation-id": "_root_node_id_0900000580004959",
        "order-number": 0,
        "binding": "CURRENT",
        "override-late-binding": true,
        "copy-behavior": 0,
        "follow-assembly": false,
        "node-id": "0000000000000000",
        "is-virtual-document": true,
        "late-binding-value": "CURRENT",
        "is-binding-broken": false,
        "are-children-compound": false,
        "is-leaf": true
      }
    }
  ]
}
```

```

    "available-versions": [
        "CURRENT",
        "1.0"
    ],
    "can-be-removed": true,
    "can-be-restructured": true,
    "child-count": 1,
    "is-compound": false,
    "is-from-assembly": false,
    "is-structurally-modified": false,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/_root_node_id_0900000580004959"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004959"
        },
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/_root_node_id_0900000580004959"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/_root_node_id_0900000580004959"
        }
    ]
},
{
    "id": "0900000580004964",
    "title": "node-1.1",
    "summary": "1",
    "updated": "2017-04-24T02:11:50.000+00:00",
    "published": "2017-04-24T02:11:50.000+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010b"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004964"
        }
    ],
    "content": {
        "name": "vd-node",
        "selected-object-name": "node-1.1",
        "selected-object-id": "0900000580004964",
        "parent-id": "0900000580004959",
        "chronicle-id": "090000058000495a",
        "relation-id": "050000058000010b",
        "order-number": 1000,
        "binding": "2.0",
        "override-late-binding": true,
        "copy-behavior": 0,
        "follow-assembly": false,
        "node-id": "0000000000000001",
        "vdm-number": "1",
        "is-virtual-document": false,
        "late-binding-value": "2.0",
        "is-binding-broken": false,
        "are-children-compound": false,

```

```
        "available-versions": [
            "CURRENT",
            "3.0",
            "2.0",
            "1.0"
        ],
        "can-be-removed": true,
        "can-be-restructured": true,
        "child-count": 0,
        "is-compound": false,
        "is-from-assembly": false,
        "is-structurally-modified": false,
        "links": [
            {
                "rel": "self",
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010b"
            },
            {
                "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/090000058000495a"
            },
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010b"
            },
            {
                "rel": "http://identifiers.emc.com/linkrel/delete",
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010b"
            }
        ]
    }
}
```



#### 2.81.2.4.3 Add a child node to a virtual document

You can use this operation when you want to add a new child node to a SysObject that is either a virtual document or a simple document. When the SysObject is a simple document, using this operation converts it into a virtual document after a child node is added to it.

Converting a simple document into virtual document is considered a special usage for adding a node. When you want to convert a simple document into a childless virtual document, you can submit an empty Request body using the POST method. After the Request has successfully completed, the virtual document has only a root node.

#### 2.81.2.4.3.1 HTTP Request method and query parameters

POST

#### 2.81.2.4.3.2 Request headers

Authorization,

Accept

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.81.2.4.3.3 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.81.2.4.3.4 Request body

When adding a node, the `<order-number>` attribute takes effect only when the new node is not the last child node. Otherwise, Foundation Java API automatically assigns a value for the `<order-number>`. For example, if there are two nodes with `<order-numbers>` 1000 and 2000; and a new node with `<order-number>` 3500 is added, the resulting `<order-number>` for the new node will be 3000, not 3500.

Similarly, when adding the first node to a virtual document; the `<order-number>` is automatically set and will always be 1000.

#### Example 2-377: XML request

```
<dm:vd-node xmlns:dm="http://identifiers.emc.com/vocab/documentum">
  <dm:chronicle-id>0900000580002516</dm:chronicle-id>
  <dm:order-number>3000</dm:order-number>
  <dm:binding>CURRENT</dm:binding>
  <dm:override-late-binding>true</dm:override-late-binding>
  <dm:copy-behavior>0</dm:copy-behavior>
  <dm:follow-assembly>false</dm:follow-assembly>
</dm:vd-node>
```



#### Example 2-378: JSON request

```
{
  "binding": "CURRENT",
  "copy-behavior": 0,
  "follow-assembly": false,
  "order-number": 3000,
  "override-late-binding": true,
  "chronicle-id": "0900000580002516"
}
```



#### 2.81.2.4.3.5 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, etc.

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.81.2.4.3.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.81.2.4.3.7 Response media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.81.2.4.3.8 Response body

##### ➡ Example 2-379: XML response

```
<?xml version='1.0' encoding='utf-8'?>
<vd-node xmlns="http://identifiers.emc.com/vocab/documentum"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <parent-id>09000005800048b3</parent-id>
    <chronicle-id>0900000580002516</chronicle-id>
    <relation-id>050000058000010b</relation-id>
    <order-number>3000</order-number>
    <binding>CURRENT</binding>
    <override-late-binding>true</override-late-binding>
    <copy-behavior>0</copy-behavior>
    <follow-assembly>false</follow-assembly>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                  050000058000010b" />
        <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
              href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
                  0900000580002516" />
        <link rel="edit"
              href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                  050000058000010b" />
        <link rel="http://identifiers.emc.com/linkrel/delete"
              href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
                  050000058000010a" />
    </links>
</vd-node>
```



##### ➡ Example 2-380: JSON response

```
{ "name": "vd-node",
  "parent-id": "09000005800048b3",
  "chronicle-id": "0900000580002516",
  "relation-id": "050000058000010a",
```

```

    "order-number": 3000,
    "binding": "CURRENT",
    "override-late-binding": true,
    "copy-behavior": 0,
    "follow-assembly": false,
    "links": [
      {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          050000058000010a"
      },
      {
        "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
          0900000580002516"
      },
      {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          050000058000010a"
      },
      {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
          050000058000010a"
      }
    ]
  }
}

```



## 2.82 Virtual Document Snapshot(s)

### 2.82.1 Virtual Document Snapshot

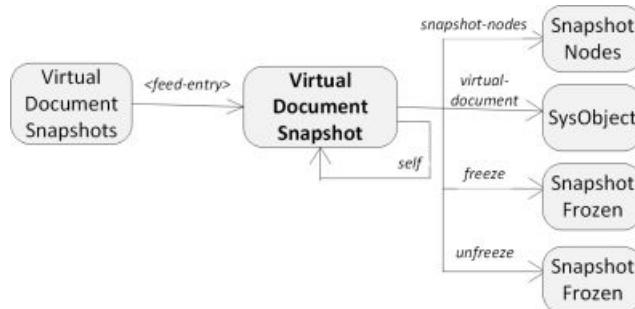
This resource represents a snapshot of a virtual document. A snapshot allows you to view the details of a virtual document at a specific point in time. You can use this resource to view the details of a snapshot or to remove a specific snapshot.



**Note:** For information on how to create a snapshot, see “[Create a snapshot](#)” on page 844.

#### 2.82.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Virtual Document Snapshots” on page 839 and About the Diagrams on page 13.

### 2.82.1.2 Feed

No feed support.

### 2.82.1.3 Link relations

Link relation	Description	Resource reference
self	This object resource	“Virtual Document Snapshot” on page 833
snapshot-nodes [1]	The virtual document nodes in a snapshot	“Virtual Document Snapshot Nodes” on page 849
freeze [1]	Used to freeze a snapshot	“Virtual Document Snapshot Frozen” on page 854
unfreeze [1]	Used to unfreeze a snapshot	“Virtual Document Snapshot Frozen” on page 854
virtual-document [1]	The virtual document object that is associated with the snapshot	“SysObject” on page 692
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.82.1.4 Operations

#### 2.82.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the snapshot from a repository
DELETE	Deletes the snapshot, but leaves the existing SysObject used to create the snapshot

#### 2.82.1.4.2 Get a Snapshot

This operation retrieves the associated Snapshot of a specified virtual document.

#### 2.82.1.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

#### 2.82.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.82.1.4.2.3 Request media types

N/A

#### 2.82.1.4.2.4 Request body

N/A

#### 2.82.1.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

#### 2.82.1.4.2.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.82.1.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.82.1.4.2.8 Response body

 **Example 2-381: XML response**

```

<vd-snapshot
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="dm_sysobject"
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
dm_sysobject">
    <properties>
        <object_name>snapshot for vd_ef716ecb-e1d2-4cd2-a312-c09baf05f5a1</object_name>
        <r_object_type>dm_sysobject</r_object_type>
        <r_creation_date>2017-06-05T02:09:53.000+00:00</r_creation_date>
        <r_modify_date>2017-06-05T02:09:53.000+00:00</r_modify_date>
        <r_modifier>dmadmin</r_modifier>
        <a_is_hidden>false</a_is_hidden>
        <i_is_deleted>false</i_is_deleted>
        <a_archive>false</a_archive>
        <a_link_resolved>false</a_link_resolved>
        <i_reference_cnt>1</i_reference_cnt>
        <i_has_folder>true</i_has_folder>
        <i_folder_id>
            <item>0c00000580001914</item>
        </i_folder_id>
        <r_link_cnt>0</r_link_cnt>
        <r_link_high_cnt>0</r_link_high_cnt>
        <r_assembled_from_id>090000058000402</r_assembled_from_id>
        <r_frzn_assembly_cnt>1</r_frzn_assembly_cnt>
        <r_has_frzn_assembly>true</r_has_frzn_assembly>
        <r_is_virtual_doc>0</r_is_virtual_doc>
        <i_contents_id>0000000000000000</i_contents_id>
        <r_page_cnt>0</r_page_cnt>
        <r_content_size>0</r_content_size>
        <a_full_text>true</a_full_text>
        <i_cabinet_id>0c00000580001914</i_cabinet_id>
        <owner_name>dmadmin</owner_name>
        <owner_permit>7</owner_permit>
        <group_name>docu</group_name>
        <group_permit>5</group_permit>
        <world_permit>3</world_permit>
        <i_antecedent_id>0000000000000000</i_antecedent_id>
        <i_chronicle_id>08000005800049fa</i_chronicle_id>
        <i_latest_flag>true</i_latest_flag>
        <r_version_label>
            <item>1.0</item>
            <item>CURRENT</item>
        </r_version_label>
        <i_branch_cnt>0</i_branch_cnt>
        <i_direct_dsc>false</i_direct_dsc>
        <r_immutable_flag>true</r_immutable_flag>
        <r_frozen_flag>true</r_frozen_flag>
        <r_has_events>false</r_has_events>
        <acl_domain>dmadmin</acl_domain>
        <acl_name>dm_4500000580001e68</acl_name>
        <i_is_reference>false</i_is_reference>
        <r_creator_name>dmadmin</r_creator_name>
        <r_is_public>true</r_is_public>
        <r_policy_id>0000000000000000</r_policy_id>
        <r_resume_state>0</r_resume_state>
        <r_current_state>0</r_current_state>
        <r_alias_set_id>0000000000000000</r_alias_set_id>
        <a_is_template>false</a_is_template>
        <r_full_content_size>0.0</r_full_content_size>
        <a_is_signed>false</a_is_signed>
        <i_partition>0</i_partition>
        <i_is_replica>false</i_is_replica>
        <i_vstamp>5</i_vstamp>
        <r_object_id>08000005800049fa</r_object_id>
    </properties>
    <links>

```

```

<link rel="self"
      href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
            08000005800049fa"/>
<link rel="http://identifiers.emc.com/linkrel/virtual-document"
      href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000580004a02"/>
<link rel="http://identifiers.emc.com/linkrel/snapshot-nodes"
      href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
            08000005800049fa/vd-nodes"/>
<link rel="http://identifiers.emc.com/linkrel/unfreeze"
      href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
            08000005800049fa/frozen"/>
</links>
</vd-snapshot>
```



### Example 2-382: JSON response

```
{
  "name": "vd-snapshot",
  "type": "dm_sysobject",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject",
  "properties": {
    "object_name": "snapshot for vd ef716ecb-e1d2-4cd2-a312-c09baf05f5a1",
    "r_object_type": "dm_sysobject",
    "r_creation_date": "2017-06-05T02:09:53.000+00:00",
    "r_modify_date": "2017-06-05T02:09:53.000+00:00",
    "r_modifier": "dmadmin",
    "a_is_hidden": false,
    "i_is_deleted": false,
    "a_archive": false,
    "a_link_resolved": false,
    "i_reference_cnt": 1,
    "i_has_folder": true,
    "i_folder_id": [
      "0c00000580001914"
    ],
    "r_link_cnt": 0,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "0900000580004a02",
    "r_frzn_assembly_cnt": 1,
    "r_has_frzn_assembly": true,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0000000000000000",
    "r_page_cnt": 0,
    "r_content_size": 0,
    "a_full_text": true,
    "i_cabinet_id": "0c00000580001914",
    "owner_name": "dmadmin",
    "owner_permit": 7,
    "group_name": "docu",
    "group_permit": 5,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "08000005800049fa",
    "i_latest_flag": true,
    "r_version_label": [
      "1.0",
      "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": true,
    "r_frozen_flag": true,
    "r_has_events": false,
    "acl_domain": "dmadmin",
    "acl_name": "dm_4500000580001e68",
    "acl_value": "0000000000000000"
  }
}
```

```
        "i_is_reference": false,
        "r_creator_name": "dmadmin",
        "r_is_public": true,
        "r_policy_id": "0000000000000000",
        "r_resume_state": 0,
        "r_current_state": 0,
        "r_alias_set_id": "0000000000000000",
        "a_is_template": false,
        "r_full_content_size": 0,
        "a_is_signed": false,
        "i_partition": 0,
        "i_is_replica": false,
        "i_vstamp": 5,
        "r_object_id": "08000005800049fa"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
08000005800049fa"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/virtual-document",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
0900000580004a02"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/snapshot-nodes",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
08000005800049fa/vd-nodes"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/unfreeze",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
08000005800049fa/frozen"
        }
    ]
}
```



#### 2.82.1.4.3 Delete a snapshot

This operation deletes a snapshot that is associated with a virtual document. The delete link is not available when the snapshot is frozen.

##### 2.82.1.4.3.1 HTTP Request method and query parameters

DELETE

**2.82.1.4.3.2 Request headers**

Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.82.1.4.3.3 Request media types**

N/A

**2.82.1.4.3.4 Request body**

N/A

**2.82.1.4.3.5 Response status**

- 204 - Deleted successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.82.1.4.3.6 Response headers**

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.82.1.4.3.7 Response media types**

N/A

**2.82.1.4.3.8 Response body**

N/A

## **2.82.2 Virtual Document Snapshots**

This resource provides a list of Virtual Document Snapshots in a repository as a feed. You can use this resource to get a list of Snapshots, or to create a new Snapshot of a Virtual Document.

### 2.82.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams](#) on page 13.

### 2.82.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: <Feed\_URI>
- Feed Title: Virtual Document Snapshots
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: Virtual-document-snapshot
- Entry ID: <Entry\_URI>
- Entry Title: Object name
- Entry Updated: Modified date

### 2.82.2.3 Link relations

Link relation	Description	Resource reference
self	This Virtual Document Snapshots resource.	<a href="#">“Virtual Document Snapshots” on page 839</a>
first, last, next, previous	Pagination links	<a href="#">“Virtual Document Snapshots” on page 839</a>

### 2.82.2.4 Operations

#### 2.82.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a list of Snapshots from a repository and presents them as a feed.
POST	Creates a snapshot for a Virtual Document and associates it with an existing SysObject or a new SysObject.

#### 2.82.2.4.2 Get Snapshots

This operation retrieves a list of snapshots for a specified virtual document and presents them as a feed.

##### 2.82.2.4.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	<ul style="list-style-type: none"> <li>• inline</li> <li>• page</li> <li>• items-per-page</li> <li>• filter</li> <li>• links</li> </ul>	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

##### Example 2-383: Use a filter to get a snapshot for a specific virtual document

Here's a code sample that demonstrates how to use a filter expression to get a list of Snapshots for a specific virtual document. In the following example, the object id of the virtual document is used as the criteria to retrieve a list of Snapshots.

```
filter=r_assembled_from_id='<VIRTUAL_DOCUMENT_OBJECT_ID>'
```

##### 2.82.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

**2.82.2.4.2.3 Request media types**

N/A

**2.82.2.4.2.4 Request body**

N/A

**2.82.2.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.82.2.4.2.6 Response headers**

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.82.2.4.2.7 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

**2.82.2.4.2.8 Response body****Example 2-384: XML response**

```
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/snapshots</id>
  <title>Virtual Document Snapshots</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2017-06-07T07:47:45.102+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">100</dm:items-per-page>
  <link rel="self" href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots"/>
  <entry>
    <id>
      http://localhost:8080/dctm-rest/repositories/REPO/snapshots/08000005800049fa
    </id>
    <title>snapshot for vd ef716ecb-e1d2-4cd2-a312-c09baf05f5a1</title>
    <summary>Snapshot 08000005800049fa for Virtual Document 0900000580004a02</summary>
    <updated>2017-06-05T02:09:53.000+00:00</updated>
    <published>2017-06-05T02:09:53.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
          08000005800049fa"/>
```

```

<content type="application/vnd.emc.documentum+xml"
          src="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
               08000005800049fa"/>
</entry>
<entry>
  <id>
    http://localhost:8080/dctm-rest/repositories/REPO/snapshots/0800000580004a26
  </id>
  <title>Snapshot from Scratch</title>
  <summary>Snapshot 0800000580004a26 for Virtual Document 0900000580004a02</summary>
  <updated>2017-06-05T02:42:19.000+00:00</updated>
  <published>2017-06-05T02:42:18.000+00:00</published>
  <link rel="edit"
        href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
               0800000580004a26"/>
  <content type="application/vnd.emc.documentum+xml"
          src="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
               0800000580004a26"/>
</entry>
</feed>
```



### Example 2-385: JSON response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots",
  "title": "Virtual Document Snapshots",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2017-06-07T07:42:21.774+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
               08000005800049fa",
      "title": "snapshot for vd ef716ecb-e1d2-4cd2-a312-c09baf05f5a1",
      "summary": "Snapshot 08000005800049fa for Virtual Document 0900000580004a02",
      "updated": "2017-06-05T02:09:53.000+00:00",
      "published": "2017-06-05T02:09:53.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
                   08000005800049fa"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
               08000005800049fa"
      }
    },
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
               0800000580004a26",
      "title": "snapshot from scratch",
      "summary": "Snapshot 0800000580004a26 for Virtual Document 0900000580004a02",
    }
  ]
}
```

```

    "updated": "2017-06-05T02:42:19.000+00:00",
    "published": "2017-06-05T02:42:18.000+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
0800000580004a26"
        }
    ],
    "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
0800000580004a26"
    }
]
}

```



#### 2.82.2.4.3 Create a snapshot

This operation creates a new snapshot of a virtual document.

##### 2.82.2.4.3.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	binding-label	The symbolic version label to use to resolve late bound nodes	string	N/A	CURRENT
	include-broken	Specifies whether to display nodes with broken bindings using the CURRENT label	string	true/false	false
	frozen	Specifies whether to freeze the snapshot right after creation	string	true/false	true

#### 2.82.2.4.3.2 Request headers

- Authorization
- Accept
- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.82.2.4.3.3 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json

#### 2.82.2.4.3.4 Request body

##### ► Example 2-386: Create a new snapshot

In the request shown below, the metadata of the snapshot is used to create a new snapshot from scratch and the virtual document reference is used to associate it with a virtual document. The system creates a new object as a Snapshot.

*XML - Request media type: application/vnd.emc.documentum+xml*

```
<vd-snapshot vd-href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000058000fe6c">
  <properties>
    <object_name>snapshot for vd 2f0479c4-7bb8-4904-abb0-95a2999c6043</object_name>
    <r_object_type>dm_sysobject</r_object_type>
  </properties>
</vd-snapshot>
```

*JSON - Request media type: application/vnd.emc.documentum+json*

```
{
  "name": "vd-snapshot",
  "vd-href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
090000058000fe6c",
  "properties": {
    "object_name": "snapshot for vd 2f0479c4-7bb8-4904-abb0-95a2999c6043",
    "r_object_type": "dm_sysobject"
  }
}
```



##### ► Example 2-387: Create a snapshot from an existing SysObject

In the Request below, a snapshot is created from an existing SysObject. A reference to an existing SysObject and a virtual document are defined in the Request. The system creates a new object as a Snapshot of the specified SysObject.

*XML - Request media type: application/vnd.emc.documentum+xml*

```
<vd-snapshot vd-href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
    090000058000fe6c"
    snapshot-href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
    090000058000ff20">
</virtual-document-snapshot>
```

*JSON - Request media type: application/vnd.emc.documentum+json*

```
{
    "name": "vd-snapshot",
    "vd-href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
    090000058000fe6c",
    "snapshot-href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/
    080000058000ccd8",
}
```



#### 2.82.2.4.3.5 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.82.2.4.3.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.82.2.4.3.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.82.2.4.3.8 Response body

##### Example 2-388: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<vd-snapshot definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
    dm_sysobject" xmlns="http://identifiers.emc.com/vocab/
    documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_sysobject">
    <properties>
        <object_name>snapshot for vd 2f0479c4-7bb8-4904-abb0-95a2999c6043</object_name>
        <r_object_type>dm_sysobject</r_object_type>
        <r_creation_date>2017-05-14T09:58:35.000+00:00</r_creation_date>
        <r_modify_date>2017-05-14T09:58:40.000+00:00</r_modify_date>
        <r_modifier>dave</r_modifier>
```

```

<a_is_hidden>false</a_is_hidden>
<i_is_deleted>false</i_is_deleted>
<a_archive>false</a_archive>
<a_link_resolved>false</a_link_resolved>
<i_reference_cnt>2</i_reference_cnt>
<i_has_folder>true</i_has_folder>
<i_folder_id>
    <item>0c00000580001909</item>
    <item>0c0000058000107</item>
</i_folder_id>
<r_link_cnt>0</r_link_cnt>
<r_link_high_cnt>0</r_link_high_cnt>
<r_assembled_from_id>090000058000ccce</r_assembled_from_id>
<r_frzn_assembly_cnt>1</r_frzn_assembly_cnt>
<r_has_frzn_assembly>true</r_has_frzn_assembly>
<r_is_virtual_doc>0</r_is_virtual_doc>
<i_contents_id>00000000000000000000</i_contents_id>
<r_page_cnt>0</r_page_cnt>
<r_content_size>0</r_content_size>
<a_full_text>true</a_full_text>
<i_cabinet_id>0c00000580001909</i_cabinet_id>
<owner_name>dave</owner_name>
<owner_permit>7</owner_permit>
<group_permit>5</group_permit>
<world_permit>3</world_permit>
<i_antecedent_id>00000000000000000000</i_antecedent_id>
<i_chronicle_id>080000058000ccd8</i_chronicle_id>
<i_latest_flag>true</i_latest_flag>
<r_version_label>
    <item>1.0</item>
    <item>CURRENT</item>
</r_version_label>
<i_branch_cnt>0</i_branch_cnt>
<i_direct_dsc>false</i_direct_dsc>
<r_immutable_flag>true</r_immutable_flag>
<r_frozen_flag>true</r_frozen_flag>
<r_has_events>false</r_has_events>
<acl_domain>dave</acl_domain>
<acl_name>dm_450000058000d07</acl_name>
<i_is_reference>false</i_is_reference>
<r_creator_name>dave</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_is_template>false</a_is_template>
<r_full_content_size>0.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>3</i_vstamp>
<r_object_id>080000058000ccd8</r_object_id>
</properties>
<links>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
        080000058000ccd8" rel="self"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
        080000058000ccd8/frozen" rel="http://identifiers.emc.com/linkrel/
        unfreeze"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
        090000058000ccce"
        rel="http://identifiers.emc.com/linkrel/virtual-document"/>
</links>
</vd-snapshot>

```



 **Example 2-389: JSON response**

```
{
  "name": "vd-snapshot",
  "type": "dm_sysobject",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject",
  "properties": {
    "object_name": "snapshot for vd 2f0479c4-7bb8-4904-abb0-95a2999c6043",
    "r_object_type": "dm_sysobject",
    "r_creation_date": "2017-05-14T09:58:35.000+00:00",
    "r_modify_date": "2017-05-14T09:58:40.000+00:00",
    "r_modifier": "dave",
    "a_is_hidden": false,
    "i_is_deleted": false,
    "a_archive": false,
    "a_link_resolved": false,
    "i_reference_cnt": 2,
    "i_has_folder": true,
    "i_folder_id": [
      "0c00000580001909",
      "0c00000580000107"
    ],
    "r_link_cnt": 0,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "090000058000ccce",
    "r_frnz_assembly_cnt": 1,
    "r_has_frnz_assembly": true,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0000000000000000",
    "r_page_cnt": 0,
    "r_content_size": 0,
    "a_full_text": true,
    "i_cabinet_id": "0c00000580001909",
    "owner_name": "dave",
    "owner_permit": 7,
    "group_permit": 5,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "080000058000ccd8",
    "i_latest_flag": true,
    "r_version_label": [
      "1.0",
      "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": true,
    "r_frozen_flag": true,
    "r_has_events": false,
    "acl_domain": "dave",
    "acl_name": "dm_4500000580000d07",
    "i_is_reference": false,
    "r_creator_name": "dave",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 0.0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 3,
    "r_object_id": "080000058000ccd8"
  },
  "links": [
    {
      "rel": "self",
      "uri": "http://localhost:8080/dctm-rest/repositories/REPO/types/dm_sysobject/080000058000ccd8"
    }
  ]
}
```

```

        "href":"http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
080000058000ccd8"
},
{
  "rel":"http://identifiers.emc.com/linkrel/unfreeze",
  "href":"http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
080000058000ccd8/frozen"
},
{
  "rel":"http://identifiers.emc.com/linkrel/virtual-document",
  "href":"http://localhost:8080/dctm-rest/repositories/REPO/objects/090000058000ccce"
}
]
}

```



## 2.83 Virtual Document Snapshot Nodes

This resource represents the components hierarchy of a virtual document snapshot .

### 2.83.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Virtual Document Snapshot” on page 833 and About the Diagrams on page 13.

### 2.83.2 Feed

Is this resource a feed? Yes.

- Feed ID: <Feed\_URI>
- Feed Title: Virtual Document Nodes
- Feed Updated: Server's current time
- Supports POST: No
- Entry: Link to entry resource definition
- Entry ID: <Entry\_URI>
- Entry Title: Object name
- Entry Updated: Modified date

Link relations	Description	Resource reference
self	This resource	“Virtual Document Snapshot Nodes” on page 849

Link relations	Description	Resource reference
first, last, next, previous	Pagination links	"Virtual Document Snapshot Nodes" on page 849

## 2.83.3 Operations

### 2.83.3.1 Supported HTTP methods

Method	Description
GET	Retrieves the virtual document nodes hierarchy of a specified snapshot object.

### 2.83.3.2 Get snapshot nodes

This operation retrieves the virtual document nodes hierarchy of a specified snapshot object.

#### 2.83.3.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	• inline • page • items-per-page • filter • links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			
	depth	To what depth level in the virtual document hierarchy that is returned.  A value of -1 returns the whole virtual document tree.	int	-1 or any integer >=0	1
	vdm-number	The node's number in the virtual document tree. The schema of <code>&lt;vdm-number&gt;</code> is based on the level of the node, for example: <ul style="list-style-type: none"><li>• 1st level: 1, 2, 3, ...</li><li>• 2nd level: 1.1, 1.2, 1.3, ...</li><li>• 3rd level: 1.1.1, 1.1.2, 1.1.3, ...</li><li>• ...</li></ul>	string	N/A	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	object-id	<p>The object id of the virtual document component of the current virtual document node.</p> <p>When the <i>&lt;object-id&gt;</i> parameter is set, the properties of the virtual document component, defined by the <i>&lt;object-id&gt;</i>, is returned.</p> <p>When the <i>&lt;object-id&gt;</i> parameter is not specified, the properties of the CURRENT version virtual document component are returned.</p>	string	N/A	N/A

#### 2.83.3.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.83.3.2.3 Request media types

N/A

#### 2.83.3.2.4 Request body

N/A

#### 2.83.3.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.83.3.2.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.83.3.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

### 2.83.3.2.8 Response body

#### Example 2-390: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/snapshots/0900000580004959/
    vd-nodes</id>
  <title>Virtual Document Nodes</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2017-04-24T02:14:18.901+00:00</updated>
  <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
  <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
  </dm:items-per-page>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/snapshots/
          0900000580004959/vd-nodes"/>
  <entry>
    <id>0900000580004959</id>
    <title>node-0</title>
    <updated>2017-04-24T02:11:51.000+00:00</updated>
    <published>2017-04-24T02:11:49.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
            _root_node_id_0900000580004959"/>
    <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
          href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000580004959"/>
    <content type="application/vnd.emc.documentum+xml"
             src="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
               _root_node_id_0900000580004959"/>
  </entry>
  <entry>
    <id>0900000580004964</id>
    <title>node-1.1</title>
    <summary>1</summary>
    <updated>2017-04-24T02:11:50.000+00:00</updated>
    <published>2017-04-24T02:11:50.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
            050000058000010b"/>
    <link rel="http://identifiers.emc.com/linkrel/virtual-document-component"
          href="http://localhost:8080/dctm-rest/repositories/REPO/objects/
            0900000580004964"/>
```

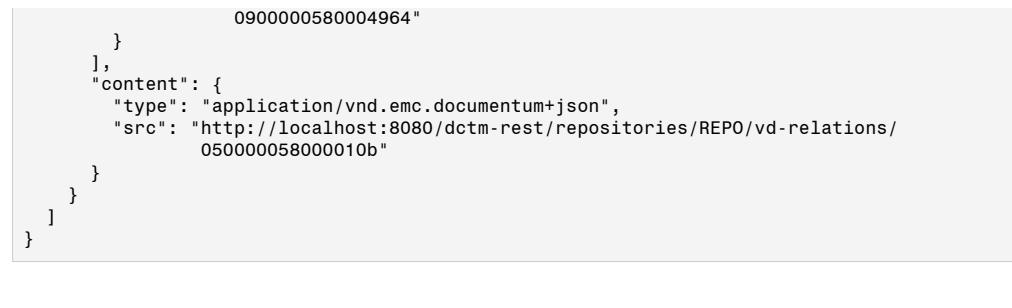
```

<content type="application/vnd.emc.documentum+xml"
src="http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/
050000058000010b"/>
</entry>
</feed>
```



### Example 2-391: JSON response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/0900000580004959/vd-nodes",
  "title": "Virtual Document Nodes",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2017-04-24T02:13:33.010+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/snapshots/0900000580004959/vd-nodes"
    }
  ],
  "entries": [
    {
      "id": "0900000580004959",
      "title": "node-0",
      "updated": "2017-04-24T02:11:51.000+00:00",
      "published": "2017-04-24T02:11:49.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/_root_node_id_0900000580004959"
        },
        {
          "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004959"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/_root_node_id_0900000580004959"
      }
    },
    {
      "id": "0900000580004964",
      "title": "node-1.1",
      "summary": "1",
      "updated": "2017-04-24T02:11:50.000+00:00",
      "published": "2017-04-24T02:11:50.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/vd-relations/050000058000010b"
        },
        {
          "rel": "http://identifiers.emc.com/linkrel/virtual-document-component",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/"
        }
      ]
    }
  ]
}
```

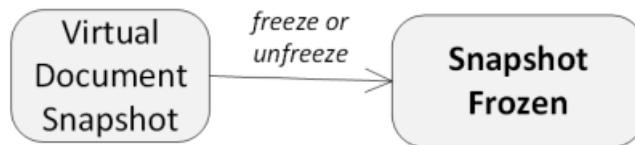


## 2.84 Virtual Document Snapshot Frozen

This resource allows you to freeze an unfrozen snapshot or unfreeze a frozen snapshot.

### 2.84.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Virtual Document Snapshot” on page 833 and About the Diagrams on page 13.

### 2.84.2 Feed

No feed support.

### 2.84.3 Operations

#### 2.84.3.1 Supported HTTP methods

Method	Description
PUT	Freeze an unfrozen snapshot.
DELETE	Unfreezes a frozen snapshot.

### 2.84.3.2 Freeze a snapshot

This operation freezes an unfrozen snapshot. An attempt to freeze a snapshot that is already frozen will cause a 409 error to occur.

A frozen snapshot is a document that has been explicitly marked as immutable. You cannot modify the content or properties of a frozen snapshot, nor can you add or remove any snapshot components. The contents and attributes of objects that are included in a frozen snapshot cannot be modified and these objects cannot be destroyed.

#### 2.84.3.2.1 HTTP Request method and query parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	N/A	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.84.3.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.84.3.2.3 Request media types

N/A

#### 2.84.3.2.4 Request body

N/A

#### 2.84.3.2.5 Response status

- 200 - Freeze of the snapshot was successful
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.84.3.2.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.84.3.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json

#### 2.84.3.2.8 Response body

When the freeze operation is successful, the response body contains a frozen snapshot object. There is an unfreeze link in the response snapshot object that allows the client to unfreeze this snapshot. Since a frozen snapshot cannot be removed, there is no delete link.

##### ➡ Example 2-392: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<vd-snapshot
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_sysobject"
    definition="http://localhost:8080/dctm-rest/repositories/ubuntudb/types/
dm_sysobject">
    <properties>
        <object_name>TEST-SNAPSHOT-c5c2d13d-dfe5-43a3-bd6d-b485504f0d7b</object_name>
        <r_object_type>dm_sysobject</r_object_type>
        <r_creation_date>2017-06-02T08:52:15.000+00:00</r_creation_date>
        <r_modify_date>2017-06-02T08:57:57.000+00:00</r_modify_date>
        <r_modifier>dmadmin</r_modifier>
        <a_is_hidden>false</a_is_hidden>
        <a_is_deleted>false</a_is_deleted>
        <a_archive>false</a_archive>
        <a_link_resolved>false</a_link_resolved>
        <i_reference_cnt>1</i_reference_cnt>
        <i_has_folder>true</i_has_folder>
        <i_folder_id>
            <item>0c00a79e80002128</item>
        </i_folder_id>
        <r_link_cnt>0</r_link_cnt>
        <r_link_high_cnt>0</r_link_high_cnt>
        <r_assembled_from_id>0900a79e80002407</r_assembled_from_id>
        <r_frzn_assembly_cnt>1</r_frzn_assembly_cnt>
        <r_has_frzn_assembly>true</r_has_frzn_assembly>
        <r_is_virtual_doc>0</r_is_virtual_doc>
        <i_contents_id>0000000000000000</i_contents_id>
        <r_page_cnt>0</r_page_cnt>
        <r_content_size>0</r_content_size>
        <a_full_text>true</a_full_text>
        <i_cabinet_id>0c00a79e80002128</i_cabinet_id>
        <owner_name>dave</owner_name>
        <owner_permit>7</owner_permit>
        <group_permit>5</group_permit>
        <world_permit>3</world_permit>
```

```

<i_antecedent_id>0000000000000000</i_antecedent_id>
<i_chronicle_id>0800a79e800022f7</i_chronicle_id>
<i_latest_flag>true</i_latest_flag>
<r_version_label>
  <item>1.0</item>
  <item>CURRENT</item>
</r_version_label>
<i_branch_cnt>0</i_branch_cnt>
<i_direct_dsc>false</i_direct_dsc>
<r_immutable_flag>true</r_immutable_flag>
<r_frozen_flag>true</r_frozen_flag>
<r_has_events>false</r_has_events>
<acl_domain>dave</acl_domain>
<acl_name>dm_4500a79e80000916</acl_name>
<i_is_reference>false</i_is_reference>
<r_creator_name>dave</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_is_template>false</a_is_template>
<r_full_content_size>0.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>5</i_vstamp>
<r_object_id>0800a79e800022f7</r_object_id>
</properties>
<links>
  <link rel="self"
    href="http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/
      0800a79e800022f7"/>
  <link rel="canonical"
    href="http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/
      0800a79e800022f7"/>
  <link rel="http://identifiers.emc.com/linkrel/virtual-document"
    href="http://localhost:8080/dctm-rest/repositories/ubuntudb/objects/
      0900a79e80002407"/>
  <link rel="http://identifiers.emc.com/linkrel/unfreeze"
    href="http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/
      0800a79e800022f7/frozen"/>
</links>
</vd-snapshot>

```



### Example 2-393: JSON response

```
{
  "name": "vd-snapshot",
  "type": "dm_sysobject",
  "definition": "http://localhost:8080/dctm-rest/repositories/ubuntudb/types/
dm_sysobject",
  "properties": {
    "object_name": "TEST-SNAPSHOT-c5c2d13d-dfe5-43a3-bd6d-b485504f0d7b",
    "r_object_type": "dm_sysobject",
    "r_creation_date": "2017-06-02T08:52:15.000+00:00",
    "r_modify_date": "2017-06-02T08:58:24.000+00:00",
    "r_modifier": "dmadmin",
    "a_is_hidden": false,
    "i_is_deleted": false,
    "a_archive": false,
    "a_link_resolved": false,
    "i_reference_cnt": 1,
    "i_has_folder": true,
    "i_folder_id": [
      "0c00a79e80002128"
    ]
  }
}
```

```
        ],
        "r_link_cnt": 0,
        "r_link_high_cnt": 0,
        "r_assembled_from_id": "0900a79e80002407",
        "r_frzn_assembly_cnt": 1,
        "r_has_frzn_assembly": true,
        "r_is_virtual_doc": 0,
        "i_contents_id": "00000000000000000000",
        "r_page_cnt": 0,
        "r_content_size": 0,
        "a_full_text": true,
        "i_cabinet_id": "0c00a79e80002128",
        "owner_name": "dave",
        "owner_permit": 7,
        "group_permit": 5,
        "world_permit": 3,
        "i_antecedent_id": "0000000000000000",
        "i_chronicle_id": "0800a79e800022f7",
        "i_latest_flag": true,
        "r_version_label": [
            "1.0",
            "CURRENT"
        ],
        "i_branch_cnt": 0,
        "i_direct_dsc": false,
        "r_immutable_flag": true,
        "r_frozen_flag": true,
        "r_has_events": false,
        "acl_domain": "dave",
        "acl_name": "dm_4500a79e80000916",
        "i_is_reference": false,
        "r_creator_name": "dave",
        "r_is_public": true,
        "r_policy_id": "0000000000000000",
        "r_resume_state": 0,
        "r_current_state": 0,
        "r_alias_set_id": "0000000000000000",
        "a_is_template": false,
        "r_full_content_size": 0,
        "a_is_signed": false,
        "i_partition": 0,
        "i_is_replica": false,
        "i_vstamp": 7,
        "r_object_id": "0800a79e800022f7"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7"
        },
        {
            "rel": "canonical",
            "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/virtual-document",
            "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/objects/0900a79e80002407"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/unfreeze",
            "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7/frozen"
        }
    ]
}
```



### 2.84.3.2.9 More samples

#### ➤ Example 2-394: An XML request

```
PUT /dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7/frozen HTTP/1.1
Host: localhost:8080
Accept: application/vnd.emc.documentum+xml
```



#### ➤ Example 2-395: An XML response

```
Status: 200
Content-Type: application/vnd.emc.documentum+xml; charset=UTF-8

<?xml version='1.0' encoding='UTF-8'?>
<vd-snapshot
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_sysobject"
    definition="http://localhost:8080/dctm-rest/repositories/ubuntudb/types/
dm_sysobject">
    <properties>
        <object_name>TEST-SNAPSHOT-c5c2d13d-dfe5-43a3-bd6d-b485504f0d7b</object_name>
        <r_object_type>dm_sysobject</r_object_type>
        <r_creation_date>2017-06-02T08:52:15.000+00:00</r_creation_date>
        <r_modify_date>2017-06-02T08:57:57.000+00:00</r_modify_date>
        <r_modifier>dadmin</r_modifier>
        <a_is_hidden>false</a_is_hidden>
        <i_is_deleted>false</i_is_deleted>
        <a_archive>false</a_archive>
        <a_link_resolved>false</a_link_resolved>
        <i_reference_cnt>1</i_reference_cnt>
        <i_has_folder>true</i_has_folder>
        <i_folder_id>
            <item>0c00a79e80002128</item>
        </i_folder_id>
        <r_link_cnt>0</r_link_cnt>
        <r_link_high_cnt>0</r_link_high_cnt>
        <r_assembled_from_id>0900a79e80002407</r_assembled_from_id>
        <r_frzn_assembly_cnt>1</r_frzn_assembly_cnt>
        <r_has_frzn_assembly>true</r_has_frzn_assembly>
        <r_is_virtual_doc>0</r_is_virtual_doc>
        <i_contents_id>0000000000000000</i_contents_id>
        <r_page_cnt>0</r_page_cnt>
        <r_content_size>0</r_content_size>
        <a_full_text>true</a_full_text>
        <i_cabinet_id>0c00a79e80002128</i_cabinet_id>
        <owner_name>dave</owner_name>
        <owner_permit>7</owner_permit>
        <group_permit>5</group_permit>
        <world_permit>3</world_permit>
        <i_antecedent_id>0000000000000000</i_antecedent_id>
        <i_chronicle_id>0800a79e800022f7</i_chronicle_id>
        <i_latest_flag>true</i_latest_flag>
        <r_version_label>
            <item>1.0</item>
            <item>CURRENT</item>
        </r_version_label>
        <i_branch_cnt>0</i_branch_cnt>
        <i_direct_dsc>false</i_direct_dsc>
        <r_immutable_flag>true</r_immutable_flag>
        <r_frozen_flag>true</r_frozen_flag>
```

```

<r_has_events>false</r_has_events>
<acl_domain>dave</acl_domain>
<acl_name>dm_4500a79e80000916</acl_name>
<i_is_reference>false</i_is_reference>
<r_creator_name>dave</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_is_template>false</a_is_template>
<r_full_content_size>0.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>5</i_vstamp>
<r_object_id>0800a79e800022f7</r_object_id>
</properties>
<links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/
            0800a79e800022f7"/>
    <link rel="canonical"
        href="http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/
            0800a79e800022f7"/>
    <link rel="http://identifiers.emc.com/linkrel/virtual-document"
        href="http://localhost:8080/dctm-rest/repositories/ubuntudb/objects/
            0900a79e80002407"/>
    <link rel="http://identifiers.emc.com/linkrel/unfreeze"
        href="http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/
            0800a79e800022f7/frozen"/>
</links>
</vd-snapshot>

```



### Example 2-396: A JSON request

```

PUT /dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7/frozen HTTP/1.1
Host: localhost:8080
Accept: application/vnd.emc.documentum+json

```



### Example 2-397: A JSON response

```

Status: 200
Content-Type: application/vnd.emc.documentum+json; charset=UTF-8

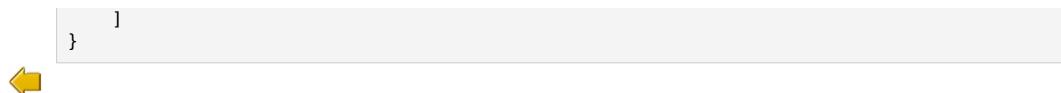
{
    "name": "vd-snapshot",
    "type": "dm_sysobject",
    "definition": "http://localhost:8080/dctm-rest/repositories/ubuntudb/types/
        dm_sysobject",
    "properties": {
        "object_name": "TEST-SNAPSHOT-c5c2d13d-dfe5-43a3-bd6d-b485504f0d7b",
        "r_object_type": "dm_sysobject",
        "r_creation_date": "2017-06-02T08:52:15.000+00:00",
        "r_modify_date": "2017-06-02T08:58:24.000+00:00",
        "r_modifier": "dmadmin",
        "a_is_hidden": false,
        "i_is_deleted": false,
        "a_archive": false,
        "a_link_resolved": false,
        "i_reference_cnt": 1,
        "i_has_folder": true,
    }
}

```

```

    "i_folder_id": [
        "0c00a79e80002128"
    ],
    "r_link_cnt": 0,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "0900a79e80002407",
    "r_frzn_assembly_cnt": 1,
    "r_has_frzn_assembly": true,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0000000000000000",
    "r_page_cnt": 0,
    "r_content_size": 0,
    "a_full_text": true,
    "i_cabinet_id": "0c00a79e80002128",
    "owner_name": "dave",
    "owner_permit": 7,
    "group_permit": 5,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "0800a79e800022f7",
    "i_latest_flag": true,
    "r_version_label": [
        "1.0",
        "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": true,
    "r_frozen_flag": true,
    "r_has_events": false,
    "acl_domain": "dave",
    "acl_name": "dm_4500a79e80000916",
    "i_is_reference": false,
    "r_creator_name": "dave",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 7,
    "r_object_id": "0800a79e800022f7"
},
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7"
    },
    {
        "rel": "canonical",
        "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/virtual-document",
        "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/objects/0900a79e80002407"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/unfreeze",
        "href": "http://localhost:8080/dctm-rest/repositories/ubuntudb/snapshots/0800a79e800022f7/frozen"
    }
]

```



### 2.84.3.3 Unfreeze a snapshot

This operation unfreezes a snapshot. Unfreezing a snapshot makes the snapshot changeable again.



**Note:** An attempt to unfreeze a snapshot that is already unfrozen will cause a 409 error to occur.

#### 2.84.3.3.1 HTTP Request method and query parameters

DELETE

#### 2.84.3.3.2 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.84.3.3.3 Request media types

N/A

#### 2.84.3.3.4 Request body

N/A

#### 2.84.3.3.5 Response status

- 204 - Unfrozen successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.84.3.3.6 Response headers

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.84.3.3.7 Response media types

N/A

### 2.84.3.3.8 Response body

N/A

## 2.84.3.4 More samples

### ➤ Example 2-398: Request

```
DELETE /dctm-rest/repositories/ubuntuDB/snapshots/0800a79e800022f7/frozen HTTP/1.1
Host: localhost:8080
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```



### ➤ Example 2-399: Response

```
Status: 204
```

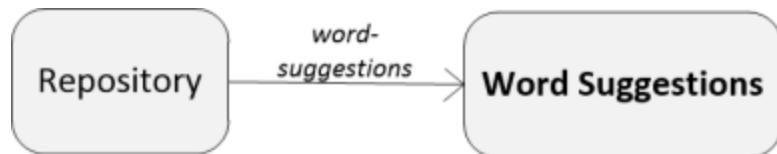


## 2.85 Word Suggestions

The *Word Suggestions* resource is part of a suite of services that allows you to use the Documentum Contextual Content Engine. This particular resource is used to suggest words that can be used in searches. It suggests words that are pronounced similarly but have different meanings, known as Homophones. It also suggests words by using auto-complete or auto-suggest, also known as Typeahead, functionality. In each of the above cases, this resource makes word suggestions that help users get the highest and best use of the Documentum Contextual Content Engine to search for content.

### 2.85.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams](#) on page 13.

## 2.85.2 Feed

Is this resource a feed? Yes.

- Feed ID: Resource URI
- Feed Title: Word Suggestions from CCE
- Feed Updated: Server's current time
- Supports POST: No
- Entry: No Resource
- Entry ID: The resource URI with a word as a query parameter. For example {repoUri}/cce/suggestions?word=xxx
- Entry Title: The word suggestion from CCE
- Entry Updated: Current server time

## 2.85.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">"Word Suggestions" on page 863</a>
search [1]	A template link formed by the search keyword	<a href="#">"Search" on page 645</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string:  
<http://identifiers.emc.com/linkrel/>.

## 2.85.4 Operations

### 2.85.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the word suggestions for requested words.

### 2.85.4.2 Get Word Suggestions

This operation retrieves word suggestions for requested words.

#### 2.85.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			
word	The word that you want to get content suggestions for	Any string	String		null
suggestion-view	Determines whether to show all suggestions, only homophones suggestions, or just typeahead suggestions	all, homophone, typeahead	String	typeahead	null

#### 2.85.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.85.4.2.3 Request media types

N/A

#### 2.85.4.2.4 Request body

N/A

#### 2.85.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.85.4.2.6 Response headers

- Content-Type
- Location

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.85.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.85.4.2.8 Response body

##### Example 2-400: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance">
<id>http://localhost:8080/dctm-rest/repositories/REPO/cce/suggestions</id>
<title>Word Suggestion from CCE</title>
<author>
    <name>OpenText Documentum</name>
</author>
<updated>2018-08-10T08:24:04.374+00:00</updated>
<dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
<dm:items-per-page
    xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    100
</dm:items-per-page>
<link rel="self"
    href="http://localhost:8080/dctm-rest/repositories/REPO/
        cce/suggestions?word=manager"></link>
<link rel="http://identifiers.emc.com/linkrel/search"
    hreftemplate="http://localhost:8080/dctm-rest/repositories/REPO/
        search?collections,facet,inline,items-per-page,
        locations,object-type,page,q,sort,timezone,view">
</link>
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/cce/suggestions?
        word=manager</id>
    <title>managers</title>
    <summary>Type Ahead Service Result</summary>
    <updated>2018-08-10T08:24:04.375+00:00</updated>
    <published>2018-08-10T08:24:04.375+00:00</published>
```

```

<link rel="http://identifiers.emc.com/linkrel/search"
      href="http://localhost:8080/dctm-rest/repositories/
          REPO/search?q=managers"></link>
<content type="application/vnd.emc.documentum+xml"
          src="http://localhost:8080/dctm-rest/repositories/REPO/cce/
              suggestions?word=manager"></content>
</entry>
<entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/cce/suggestions?
        word=manager</id>
    <title>manager job</title>
    <summary>Type Ahead Service Result</summary>
    <updated>2018-08-10T08:24:04.375+00:00</updated>
    <published>2018-08-10T08:24:04.375+00:00</published>
    <link rel="http://identifiers.emc.com/linkrel/search"
          href="http://localhost:8080/dctm-rest/repositories/REPO/search?
              q=manager%20job"></link>
    <content type="application/vnd.emc.documentum+xml"
              src="http://localhost:8080/dctm-rest/repositories/REPO/cce/
                  suggestions?word=manager"></content>
</entry>
</feed>
```



#### Example 2-401: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/suggestions",
    "title": "Word Suggestion from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-08-10T08:15:12.160+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                suggestions?word=manager"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/search",
            "hreftemplate": "http://localhost:8080/dctm-rest/repositories/
                REPO/search?{collections,facet,inline,
                items-per-page,locations,object-type,page,q,sort,
                timezone,view}"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                suggestions?word=manager",
            "title": "managers",
            "summary": "Type Ahead Service Result ",
            "updated": "2018-08-10T08:15:12.166+00:00",
            "published": "2018-08-10T08:15:12.166+00:00",
            "links": [
                {
                    "rel": "http://identifiers.emc.com/linkrel/search",
                    "href": "http://localhost:8080/dctm-rest/repositories/
                        REPO/search?q=managers"
                }
            ],
            "content": {
```

```

        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
               suggestions?word=manager"
    }
},
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
          suggestions?word=manager",
    "title": "manager job",
    "summary": "Type Ahead Service Result ",
    "updated": "2018-08-10T08:15:12.167+00:00",
    "published": "2018-08-10T08:15:12.167+00:00",
    "links": [
        {
            "rel": "http://identifiers.emc.com/linkrel/search",
            "href": "http://localhost:8080/dctm-rest/repositories/
                     REPO/search?q=manager%20job"
        }
    ],
    "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
               suggestions?word=manager"
    }
}
]
}

```



### ➡ Example 2-402: HAL: JSON Response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/suggestions",
    "title": "Word Suggestion from CCE",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2018-08-10T08:24:42.101+00:00",
    "page": 1,
    "items-per-page": 100,
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                       suggestions?word=manager",
                "title": "managers",
                "summary": "Type Ahead Service Result ",
                "updated": "2018-08-10T08:24:42.102+00:00",
                "published": "2018-08-10T08:24:42.102+00:00",
                "links": {
                    "http://identifiers.emc.com/linkrel/search": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
                                 REPO/search?q=managers"
                    },
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/
                                 REPO/cce/suggestions?word=manager"
                    }
                }
            },
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
                      suggestions?word=manager",
                "title": "manager job",
                "summary": "Type Ahead Service Result "
            }
        ]
    }
}
```

```

    "updated": "2018-08-10T08:24:42.102+00:00",
    "published": "2018-08-10T08:24:42.102+00:00",
    "_links": {
        "http://identifiers.emc.com/linkrel/search": {
            "href": "http://localhost:8080/dctm-rest/repositories/
REPO/search?q=manager%20job"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
REPO/cce/suggestions?word=manager"
        }
    }
},
"_links": {
    "http://identifiers.emc.com/linkrel/search": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
search{?collections,facet,inline,
items-per-page,locations,object-type,page,q,sort,
timezone,view}",
        "templated": true
    },
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/cce/
suggestions?word=manager"
    }
}
}

```



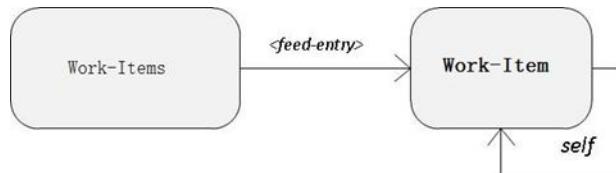
## 2.86 Work Item(s)

### 2.86.1 Work Item

The *Work Item* resource is used to get a specified work item.

#### 2.86.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Work Items](#)” on page 874 and [About the Diagrams](#) on page 13.

### 2.86.1.2 Feed

No feed support.

### 2.86.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">"Work Item" on page 869</a>
workflow [1]	A workflow object	<a href="#">"Workflow" on page 906</a>
workflow-activity [1]	A workflow-activity object	<a href="#">"Workflow Activity" on page 935</a>

[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <http://identifiers.emc.com/linkrel/>.

### 2.86.1.4 Operations

#### 2.86.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a work item

#### 2.86.1.4.2 Get Work Item

This operation retrieves a Work Item object.

##### 2.86.1.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	view				

##### 2.86.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.86.1.4.2.3 Request media types**

N/A

**2.86.1.4.2.4 Request body**

N/A

**2.86.1.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.86.1.4.2.6 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.86.1.4.2.7 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

**2.86.1.4.2.8 Response body**

 **Example 2-403: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<work-item
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dmi_workitem"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dmi_workitem">
<properties>
    <r_workflow_id>4d00000180000100</r_workflow_id>
    <r_act_seqno>4</r_act_seqno>
    <r_performer_name>dadmin</r_performer_name>
    <r_creation_date>2018-12-25T09:18:45.000+00:00</r_creation_date>
    <r_priority>0</r_priority>
    <r_auto_method_id>1000000180000268</r_auto_method_id>
    <return_value></return_value>
    <r_exec_result_id>0000000000000000</r_exec_result_id>
    <r_exec_launch>false</r_exec_launch>
    <r_exec_time_out>false</r_exec_time_out>
    <r_runtime_state>2</r_runtime_state>
    <r_queue_item_id>0000000000000000</r_queue_item_id>
    <r_act_def_id>4c000001800025c4</r_act_def_id>
```

```

<r_sign_off_req>false</r_sign_off_req>
<r_launch_timeout>2018-12-25T09:19:45.000+00:00</r_launch_timeout>
<a_wq_name>3d00000180000102</a_wq_name>
<a_wq_flag>0</a_wq_flag>
<a_wq_policy_id>0000000000000000</a_wq_policy_id>
<user_time>0</user_time>
<user_cost>0.0</user_cost>
<r_exec_retried_count>0</r_exec_retried_count>
<r_handling_instruction>0</r_handling_instruction>
<r_target_task_id>0000000000000000</r_target_task_id>
<i_is_replica>false</i_is_replica>
<i_vstamp>4</i_vstamp>
<r_object_id>4a00000180000103</r_object_id>
</properties>
<links>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/work-items/
              4a00000180000103"
        rel="self"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
              4d00000180000100"
        rel="http://identifiers.emc.com/linkrel/workflow"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-activities/4c000001800025c4"
        rel="http://identifiers.emc.com/linkrel/workflow-activity"/>
</links>
</work-item>

```



### Example 2-404: JSON response

```
{
    "type": "dmi_workitem",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dmi_workitem",
    "properties": {
        "r_workflow_id": "4d00000180000100",
        "r_act_seqno": 4,
        "r_performer_name": "dmadmin",
        "r_creation_date": "2018-12-25T09:18:45.000+00:00",
        "r_priority": 0,
        "r_auto_method_id": "1000000180000268",
        "return_value": 0,
        "r_exec_result_id": "0000000000000000",
        "r_exec_launch": false,
        "r_exec_time_out": false,
        "r_runtime_state": 2,
        "r_queue_item_id": "0000000000000000",
        "r_act_def_id": "4c000001800025c4",
        "r_sign_off_req": false,
        "r_launch_timeout": "2018-12-25T09:19:45.000+00:00",
        "a_wq_name": "3d00000180000102",
        "a_wq_flag": 0,
        "a_wq_policy_id": "0000000000000000",
        "user_time": 0,
        "user_cost": 0.0,
        "r_exec_retried_count": 0,
        "r_handling_instruction": 0,
        "r_target_task_id": "0000000000000000",
        "i_is_replica": false,
        "i_vstamp": 4,
        "r_object_id": "4a00000180000103"
    },
    "links": [
        {
            "rel": "self",

```

```

        "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
        4a00000180000103"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180000100"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-activities/4c000001800025c4"
    }
]
}

```



#### ➡ Example 2-405: HAL: JSON Response

```

{
    "type": "dmi_workitem",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/dmi_workitem",
    "properties": {
        "r_workflow_id": "4d00000180000100",
        "r_act_seqno": 4,
        "r_performer_name": "dmadmin",
        "r_creation_date": "2018-12-25T09:18:45.000+00:00",
        "r_priority": 0,
        "r_auto_method_id": "1000000180000268",
        "return_value": 0,
        "r_exec_result_id": "0000000000000000",
        "r_exec_launch": false,
        "r_exec_time_out": false,
        "r_runtime_state": 2,
        "r_queue_item_id": "0000000000000000",
        "r_act_def_id": "4c000001800025c4",
        "r_sign_off_req": false,
        "r_launch_timeout": "2018-12-25T09:19:45.000+00:00",
        "a_wq_name": "3d00000180000102",
        "a_wq_flag": 0,
        "a_wq_policy_id": "0000000000000000",
        "user_time": 0,
        "user_cost": 0.0,
        "r_exec_retried_count": 0,
        "r_handling_instruction": 0,
        "r_target_task_id": "0000000000000000",
        "i_is_replica": false,
        "i_vstamp": 4,
        "r_object_id": "4a00000180000103"
    },
    "_links": {
        "http://identifiers.emc.com/linkrel/workflow": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d00000180000100"
        },
        "http://identifiers.emc.com/linkrel/workflow-activity": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c4"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
            4a00000180000103"
        }
    }
}

```

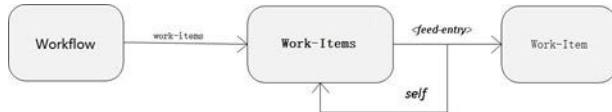


## 2.86.2 Work Items

The *Work Items* resource is used to get the Work Items of a specified Workflow instance.

### 2.86.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Workflow” on page 906 and About the Diagrams on page 13.

### 2.86.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Work items
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “Work Item” on page 869 resource
- Entry ID: The entity URI
- Entry Title: The id of the Work Item
- Entry Updated: The last modify date

### 2.86.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	“Work Items” on page 874

## 2.86.2.4 Operations

### 2.86.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the Work Items of the specified Workflow instance

### 2.86.2.4.2 Get Work Items

This operation retrieves the Work Items of the specified Workflow instance

#### 2.86.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, include-total, links, sort, view, filter, inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.86.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.86.2.4.2.3 Request media types

N/A

#### 2.86.2.4.2.4 Request body

N/A

#### 2.86.2.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.86.2.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.86.2.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.86.2.4.2.8 Response body

##### Example 2-406: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>
      http://localhost:8080/dctm-rest/repositories/REPO/workflows/
      4d00000180000100/work-items
    </id>
    <title>Work items</title>
    <author>
      <name>OpenText Documentum</name>
    </author>
    <updated>2019-02-27T02:30:14.521+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
      100
    </dm:items-per-page>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
      4d00000180000100/work-items" rel="self"/>
    <entry>
      <id>
        http://localhost:8080/dctm-rest/repositories/REPO/work-items/
        4a00000180000103
      </id>
      <title>4a00000180000103</title>
      <author>
        <name>dmadmin</name>
        <uri>
          http://localhost:8080/dctm-rest/repositories/REPO/users/dmadmin
        </uri>
      </author>
      <summary>FINISHED</summary>
      <updated>2019-02-27T02:30:14.521+00:00</updated>
      <published>2018-12-25T09:18:45.000+00:00</published>
      <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/work-items/
        4a00000180000103" rel="edit"/>
      <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180000100"
        rel="http://identifiers.emc.com/linkrel/workflow"/>
      <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
```

```

        workflow-activities/4c000001800025c4"
    rel="http://identifiers.emc.com/linkrel/workflow-activity"/>
<content src="http://localhost:8080/dctm-rest/repositories/REPO/
    work-items/4a00000180000103"
    type="application/vnd.emc.documentum+xml"/>
</entry>
</feed>

```



### Example 2-407: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180000100/work-items",
    "title": "Work items",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-27T02:30:20.412+00:00",
    "page": 1,
    "items-per-page": 100,
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d00000180000100/work-items"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-items/
                4a00000180000103",
            "title": "4a00000180000103",
            "author": [
                {
                    "name": "dmadmin",
                    "uri": "http://localhost:8080/dctm-rest/repositories/REPO/users/
                        dmadmin"
                }
            ],
            "summary": "FINISHED",
            "updated": "2019-02-27T02:30:20.412+00:00",
            "published": "2018-12-25T09:18:45.000+00:00",
            "links": [
                {
                    "rel": "edit",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                        work-items/4a00000180000103"
                },
                {
                    "rel": "http://identifiers.emc.com/linkrel/workflow",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                        workflows/4d00000180000100"
                },
                {
                    "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                        workflow-activities/4c000001800025c4"
                }
            ],
            "content": [
                {
                    "type": "application/vnd.emc.documentum+json",
                    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                        work-items/4a00000180000103"
                }
            ],
            "content": [
                {
                    "type": "application/vnd.emc.documentum+json",
                    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                        work-items/4a00000180000100"
                }
            ]
        }
    ]
}
```

```
        }
    ]
}
```



### Example 2-408: HAL: JSON Response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180000100/work-items",
  "title": "Work items",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-27T02:30:22.797+00:00",
  "page": 1,
  "items-per-page": 100,
  "_embedded": {
    "collection": [
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
          work-items/4a00000180000103",
        "title": "4a00000180000103",
        "author": [
          {
            "name": "dmadmin",
            "uri": "http://localhost:8080/dctm-rest/repositories/REPO/
              users/dmadmin"
          }
        ],
        "summary": "FINISHED",
        "updated": "2019-02-27T02:30:22.797+00:00",
        "published": "2018-12-25T09:18:45.000+00:00",
        "_links": {
          "http://identifiers.emc.com/linkrel/workflow": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              workflows/4d00000180000100"
          },
          "http://identifiers.emc.com/linkrel/workflow-activity": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-activities/4c000001800025c4"
          },
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              work-items/4a00000180000103"
          }
        }
      }
    ],
    "_links": {
      "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
          4d00000180000100/work-items"
      }
    }
}
```



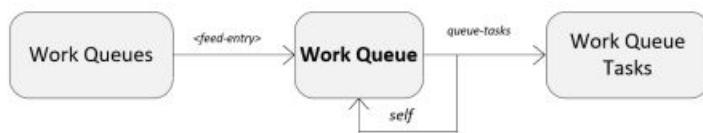
## 2.87 Work Queue(s)

### 2.87.1 Work Queue

The *Work Queue* resource is used to get a specified single work queue.

#### 2.87.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Work Queues](#)” on page 882 and [About the Diagrams](#) on page 13.

#### 2.87.1.2 Feed

No feed support.

#### 2.87.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">“Work Queue” on page 879</a>
queue-tasks [1]	Get the queue tasks under this work queue	<a href="#">“Work Queue Tasks” on page 898</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .		

#### 2.87.1.4 Operations

##### 2.87.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a Work Queue

**2.87.1.4.2 Get work queue**

This operation retrieves a Work Queue object.

**2.87.1.4.2.1 HTTP Request Method and Query Parameters**

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, views	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

**2.87.1.4.2.2 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

**2.87.1.4.2.3 Request media types**

N/A

**2.87.1.4.2.4 Request body**

N/A

**2.87.1.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

**2.87.1.4.2.6 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.87.1.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.87.1.4.2.8 Response body

 **Example 2-409: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<work-queue xmlns="http://identifiers.emc.com/vocab/documentum"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:type="dmc_workqueue" definition="http://localhost:8080/dctm-rest/repositories/REPO/
             types/dmc_workqueue">
    <properties>
        <wq_name>test work queue</wq_name>
        <wq_category_id>0b00000180008068</wq_category_id>
        <wq_policy_id>0000000180000d1d</wq_policy_id>
        <i_is_replica>false</i_is_replica>
        <i_vstamp>0</i_vstamp>
        <r_object_id>0000000180000d20</r_object_id>
    </properties>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
              work-queues/0000000180000d20"></link>
        <link rel="http://identifiers.emc.com/linkrel/queue-tasks"
              href="http://localhost:8080/dctm-rest/repositories/REPO/
              queue-tasks?queue-names=test%2Bwork%2Bqueue"></link>
    </links>
</work-queue>
```



 **Example 2-410: A request and JSON response**

This code sample shows you the Work Queue Request and the Response.

#### Request

```
GET /dctm-rest/repositories/REPO/work-queue/0000000180000d20 HTTP/1.1
Host: localhost:8080
Accept: application/vnd.emc.documentum+json
Authorization: Basic QWRtaW5pc3RyYXRvcjpQYXNzd29yZEAxMjM=
```

#### JSON Response

```
{
    "name": "work-queue",
    "type": "dmc_workqueue",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
                  dmc_workqueue",
    "properties": {
        "wq_name": "test work queue",
        "wq_category_id": "0b00000180008068",
        "wq_policy_id": "0000000180000d1d",
        "i_is_replica": false,
        "i_vstamp": 0,
```

```

        "r_object_id": "0000000180000d20"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
                    0000000180000d20"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/queue-tasks",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    queue-tasks?queue-names=test%2Bwork%2Bqueue"
        }
    ]
}

```



#### ➤ Example 2-411: HAL: JSON Response

```

{
    "name": "work-queue",
    "type": "dmc_workqueue",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
                    dmc_workqueue",
    "properties": {
        "wd_name": "test work queue",
        "wd_category_id": "0b00000180008068",
        "wd_policy_id": "0000000180000d1d",
        "i_is_replica": false,
        "i_vstamp": 0,
        "r_object_id": "0000000180000d20"
    },
    "_links": {
        "http://identifiers.emc.com/linkrel/queue-tasks": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    queue-tasks?queue-names=test%2Bwork%2Bqueue"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
                    0000000180000d20"
        }
    }
}

```



## 2.87.2 Work Queues

The *Work Queues* resource is used to get a collection of work queues to which a specific user belongs.

### 2.87.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams on page 13](#).

### 2.87.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Work queues
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “[Work Queue](#)” on page 879 resource
- Entry ID: The entity URI
- Entry Title: The name (*wq\_name*) of the Work Queue
- Entry Updated: The current time of the server

### 2.87.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	“ <a href="#">Work Queues</a> ” on page 882

### 2.87.2.4 Operations

#### 2.87.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves Work Queues

#### 2.87.2.4.2 Get work queues

This operation retrieves a list of work queues to which the specified user belongs. It retrieves, by default, the queue name of the user that is currently logged in.

##### 2.87.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	inline, inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	user-name	Specifies the name of the user	String	N/A	The user that is currently logged in

##### 2.87.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.87.2.4.2.3 Request media types

N/A

##### 2.87.2.4.2.4 Request body

N/A

##### 2.87.2.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.87.2.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.87.2.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.87.2.4.2.8 Response body

##### Example 2-412: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/work-queues</id>
  <title>Work queues</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2019-06-12T09:20:30.728+00:00</updated>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
        work-queues"></link>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
      0000000180000d2a</id>
    <title>test-work-queue-2</title>
    <summary>0000000180000d2a test-work-queue-2</summary>
    <updated>2019-06-12T09:20:30.728+00:00</updated>
    <published>2019-06-12T09:20:30.728+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
            0000000180000d2a"></link>
    <content type="application/vnd.emc.documentum+xml"
              src="http://localhost:8080/dctm-rest/repositories/REPO/
              work-queues/0000000180000d2a"></content>
  </entry>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
      0000000180000d20</id>
    <title>test work queue</title>
    <summary>0000000180000d20 test work queue</summary>
    <updated>2019-06-12T09:20:30.728+00:00</updated>
    <published>2019-06-12T09:20:30.728+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
            0000000180000d20"></link>
    <content type="application/vnd.emc.documentum+xml"
              src="http://localhost:8080/dctm-rest/repositories/REPO/
              work-queues/0000000180000d20"></content>
  </entry>
</feed>
```



 **Example 2-413: JSON response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-queues",
  "title": "Work queues",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-06-12T09:18:31.925+00:00",
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
        work-queues"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
        0000000180000d2a",
      "title": "test-work-queue-2",
      "summary": "0000000180000d2a test-work-queue-2",
      "updated": "2019-06-12T09:18:31.939+00:00",
      "published": "2019-06-12T09:18:31.939+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            work-queues/0000000180000d2a"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
          work-queues/0000000180000d2a"
      }
    },
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-queues/
        0000000180000d20",
      "title": "test work queue",
      "summary": "0000000180000d20 test work queue",
      "updated": "2019-06-12T09:18:31.939+00:00",
      "published": "2019-06-12T09:18:31.939+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO
            work-queues/0000000180000d20"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
          work-queues/0000000180000d20"
      }
    }
  ]
}
```



 **Example 2-414: HAL: JSON Response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/work-queues",
  "title": "Work queues",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
}
```

```

"updated": "2019-06-12T09:20:57.243+00:00",
"_embedded": {
    "collection": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                work-queues/0000000180000d2a",
            "title": "test-work-queue-2",
            "summary": "0000000180000d2a test-work-queue-2",
            "updated": "2019-06-12T09:20:57.243+00:00",
            "published": "2019-06-12T09:20:57.243+00:00",
            "_links": {
                "self": {
                    "href": "http://localhost:8080/dctm-rest/repositories/
                        REPO/work-queues/0000000180000d2a"
                }
            }
        },
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                work-queues/0000000180000d20",
            "title": "test work queue",
            "summary": "0000000180000d20 test work queue",
            "updated": "2019-06-12T09:20:57.243+00:00",
            "published": "2019-06-12T09:20:57.243+00:00",
            "_links": {
                "self": {
                    "href": "http://localhost:8080/dctm-rest/repositories/
                        REPO/work-queues/0000000180000d20"
                }
            }
        }
    ],
    "_links": {
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                work-queues"
        }
    }
}

```



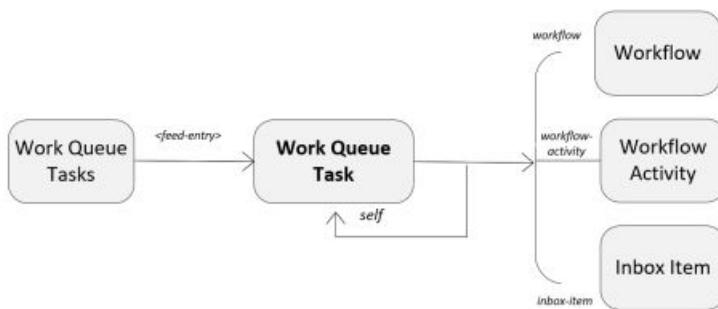
## 2.88 Work Queue Task(s)

### 2.88.1 Work Queue Task

The *Work Queue Task* resource is used to retrieve a queue task, which allows you to perform some actions on it.

#### 2.88.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Work Queue Tasks” on page 898 and About the Diagrams on page 13.

### 2.88.1.2 Feed

No feed support.

### 2.88.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	“Work Queue Task” on page 887
workflow [1]	Link to the workflow that this queue task belongs to	“Workflow” on page 906
workflow-activity [1]	Link to a workflow activity	“Workflow Activity” on page 935
inbox-item [1]	Link to an inbox item	“Inbox Item” on page 343
[1]	This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <code>http://identifiers.emc.com/linkrel/</code> .	

### 2.88.1.4 Operations

#### 2.88.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a work queue task
PUT	Perform actions on a work queue task

#### 2.88.1.4.2 Get work queue task

This operation retrieves a Work Queue task.

##### 2.88.1.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

##### 2.88.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

##### 2.88.1.4.2.3 Request media types

N/A

##### 2.88.1.4.2.4 Request body

N/A

##### 2.88.1.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

##### 2.88.1.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.88.1.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.88.1.4.2.8 Response body

##### ➡ Example 2-415: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<work-queue-task xmlns="http://identifiers.emc.com/vocab/documentum"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <original-wq-name>test-work-queue-2</original-wq-name>
    <pulled>true</pulled>
    <held>false</held>
    <held-by></held-by>
    <wq-policy-id>000000018000d1d</wq-policy-id>
    <wq-doc-profile></wq-doc-profile>
    <increment-priority>1</increment-priority>
    <skill-level></skill-level>
    <process-id>4b0000018001f56c</process-id>
    <is-signoff-required>false</is-signoff-required>
    <is-repeatable>true</is-repeatable>
    <is-delegable>true</is-delegable>
    <is-end-activity>true</is-end-activity>
    <is-distribution-listtsak>false</is-distribution-listtsak>
    <is-group-owned>false</is-group-owned>
    <runtime-state>DORMANT</runtime-state>
    <is-failed-autotask>false</is-failed-autotask>
    <is-manual-transition>false</is-manual-transition>
    <instruction></instruction>
    <is-sequential-a-activity>false</is-sequential-a-activity>
    <has-self-reject-link>false</has-self-reject-link>
    <is-first-performer>false</is-first-performer>
    <is-last-performer>false</is-last-performer>
    <has-forward-activities>false</has-forward-activities>
    <has-reject-activities>false</has-reject-activities>
    <queue-item-id>1b0000018000e550</queue-item-id>
    <activity-id>4c0000018001f578</activity-id>
    <activity-name>work queue2</activity-name>
    <is-performer-assignment-required>false</is-performer-assignment-required>
    <complete-error-status>COMPLETE_UNKNOWN</complete-error-status>
    <task-name>work queue2</task-name>
    <task-type>WORKFLOWTASK</task-type>
    <task-state>DORMANT</task-state>
    <priority>171</priority>
    <sent-date>2019-06-06T05:00:29.000+00:00</sent-date>
    <item-id>4a00000180001114</item-id>
    <workflow-id>4d00000180001114</workflow-id>
    <workflow-task-type>UNKNOWN</workflow-task-type>
    <performer>Administrator</performer>
    <stamp>1b0000018000e550</stamp>
    <links>
        <link rel="self"
              href="http://localhost:8080/dctm-rest/repositories/REPO/queue-tasks/
                    4a00000180001114"></link>
        <link rel="http://identifiers.emc.com/linkrel/workflow"
              href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                    4d00000180001114"></link>
        <link rel="http://identifiers.emc.com/linkrel/workflow-activity"
              href="http://localhost:8080/dctm-rest/repositories/REPO/">
```

```

        workflow-activities/4c0000018001f578"></link>
    <link rel="http://identifiers.emc.com/linkrel/inbox-item"
          href="http://localhost:8080/dctm-rest/repositories/REPO/inbox/
              1b0000018000e550"></link>
    </links>
</work-queue-task>

```



#### Example 2-416: JSON response

```
{
    "name": "work-queue-task",
    "original-wq-name": "test-work-queue-2",
    "pulled": true,
    "held": false,
    "held-by": "",
    "wq-policy-id": "0000000180000d1d",
    "wq-doc-profile": "",
    "increment-priority": 1,
    "skill-level": 0,
    "process-id": "4b0000018001f56c",
    "is-signoff-required": false,
    "is-repeatable": true,
    "is-delegable": true,
    "is-end-activity": true,
    "is-distribution-listtsak": false,
    "is-group-owned": false,
    "runtime-state": "DORMANT",
    "is-failed-autotask": false,
    "is-manual-transition": false,
    "instruction": "",
    "is-sequential-activity": false,
    "has-self-reject-link": false,
    "is-first-performer": false,
    "is-last-performer": false,
    "has-forward-activities": false,
    "has-reject-activities": false,
    "queue-item-id": "1b0000018000e550",
    "activity-id": "4c0000018001f578",
    "activity-name": "work queue2",
    "is-performer-assignment-required": false,
    "complete-error-status": "COMPLETE_UNKNOWN",
    "task-name": "work queue2",
    "task-type": "WORKFLOWTASK",
    "task-state": "DORMANT",
    "priority": 171,
    "sent-date": "2019-06-06T05:00:29.000+00:00",
    "item-id": "4a00000180001114",
    "workflow-id": "4d00000180001114",
    "workflow-task-type": "UNKNOWN",
    "performer": "Administrator",
    "stamp": "1b0000018000e550",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                queue-tasks/4a00000180001114"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/workflow",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d00000180001114"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-activities/4c0000018001f578"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/inbox-item",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
                1b0000018000e550"
        }
    ]
}
```

```

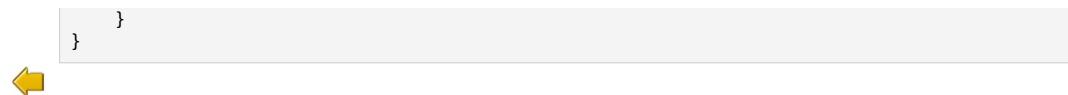
        }
    ]
}
} 1b0000018000e550"

```



### Example 2-417: HAL: JSON Response

```
{
  "name": "work-queue-task",
  "original-wq-name": "test-work-queue-2",
  "pulled": true,
  "held": false,
  "held-by": "",
  "wq-policy-id": "0000000180000d1d",
  "wq-doc-profile": "",
  "increment-priority": 1,
  "skill-level": 0,
  "process-id": "4b0000018001f56c",
  "is-signoff-required": false,
  "is-repeatable": true,
  "is-delegable": true,
  "is-end-activity": true,
  "is-distribution-listtsak": false,
  "is-group-owned": false,
  "runtime-state": "DORMANT",
  "is-failed-autotask": false,
  "is-manual-transition": false,
  "instruction": "",
  "is-sequential-a-activity": false,
  "has-self-reject-link": false,
  "is-first-performer": false,
  "is-last-performer": false,
  "has-forward-activities": false,
  "has-reject-activities": false,
  "queue-item-id": "1b0000018000e550",
  "activity-id": "4c0000018001f578",
  "activity-name": "work queue2",
  "is-performer-assignment-required": false,
  "complete-error-status": "COMPLETE_UNKNOWN",
  "task-name": "work queue2",
  "task-type": "WORKFLOWTASK",
  "task-state": "DORMANT",
  "priority": 171,
  "sent-date": "2019-06-06T05:00:29.000+00:00",
  "item-id": "4a00000180001114",
  "workflow-id": "4d00000180001114",
  "workflow-task-type": "UNKNOWN",
  "performer": "Administrator",
  "stamp": "1b0000018000e550",
  "_links": {
    "http://identifiers.emc.com/linkrel/workflow": {
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/4d00000180001114"
    },
    "http://identifiers.emc.com/linkrel/workflow-activity": {
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-activities/4c0000018001f578"
    },
    "self": {
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/queue-tasks/4a00000180001114"
    },
    "http://identifiers.emc.com/linkrel/inbox-item": {
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/1b0000018000e550"
    }
  }
}
```



#### 2.88.1.4.3 Do actions on queue task

The Work Queue Task is also a Workflow Task. In addition to supporting the actions on a workflow task such as acquire, forward, reject, delegate, repeat, signoff, assign and complete through the Inbox Item resource, it also supports the actions, shown below, on the Work Queue Task resource.

##### 2.88.1.4.3.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	action	<p>The specific action to be performed on the current Work Queue Task resource. Currently, the following actions are supported:</p> <ul style="list-style-type: none"> <li>• <b>HOLD</b>: Sets the work item <i>a_hold_by</i> parameter to indicate if the task is being worked on</li> <li>• <b>REASSIGN</b>: Reassigns this task to the specified user, the <i>user-name</i> parameter is required, but the <i>pulled</i> parameter is optional. This action unsuspends a suspended task</li> <li>• <b>SUSPEND</b>: Suspends the specified task with the specified expected unsuspend date and time. The <i>unsuspend-time</i> is optional</li> <li>• <b>UNSUSPEND</b>: Unsuspends or resumes the specified task</li> <li>• <b>UNHOLD</b>: Releases the task</li> </ul>	String	N/A	N/A
	user-name	Specifies the user name of the user to whom the task is reassigned			

HTTP method	Query parameters	Description	Data type	Value range	Default value
	pulled	It is <i>true</i> to indicate when an end user assigns the task, or <i>false</i> if a supervisor or administrator assigns the task	Boolean	N/A	<i>false</i>
	unsuspend-time	Specifies the absolute date and time when the task must be automatically unsuspended when the action is SUSPEND. Pass in NULL if the timer is not used for suspending the task. The date format should comply with ISO 8601, e.g. 2050-12-31T23:59:59+08:00	String	N/A	<i>null</i>

#### 2.88.1.4.3.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.88.1.4.3.3 Request media types

N/A

#### 2.88.1.4.3.4 Request body

N/A

#### 2.88.1.4.3.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.88.1.4.3.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.88.1.4.3.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.88.1.4.3.8 Response body

##### Example 2-418: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<work-queue-task xmlns="http://identifiers.emc.com/vocab/documentum"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <original-wq-name>test-work-queue-2</original-wq-name>
    <pulled>true</pulled>
    <held>false</held>
    <held-by></held-by>
    <wq-policy-id>0000000180000d1d</wq-policy-id>
    <wq-doc-profile></wq-doc-profile>
    <increment-priority>1</increment-priority>
    <skill-level>0</skill-level>
    <process-id>4b0000018001f56c</process-id>
    <is-signoff-required>false</is-signoff-required>
    <is-repeatable>true</is-repeatable>
    <is-delegable>true</is-delegable>
    <is-end-activity>true</is-end-activity>
    <is-distribution-listtsak>false</is-distribution-listtsak>
    <is-group-owned>false</is-group-owned>
    <runtime-state>DORMANT</runtime-state>
    <is-failed-autotask>false</is-failed-autotask>
    <is-manual-transition>false</is-manual-transition>
    <instruction></instruction>
    <is-sequential-activity>false</is-sequential-activity>
    <has-self-reject-link>false</has-self-reject-link>
    <is-first-performer>false</is-first-performer>
    <is-last-performer>false</is-last-performer>
    <has-forward-activities>false</has-forward-activities>
    <has-reject-activities>false</has-reject-activities>
    <queue-item-id>1b0000018000550</queue-item-id>
    <activity-id>4c0000018001f578</activity-id>
    <activity-name>work queue2</activity-name>
    <is-performer-assignment-required>false</is-performer-assignment-required>
    <complete-error-status>COMPLETE_UNKNOWN</complete-error-status>
    <task-name>work queue2</task-name>
    <task-type>WORKFLOWTASK</task-type>
    <task-state>DORMANT</task-state>
    <priority>171</priority>
    <sent-date>2019-06-06T05:00:29.000+00:00</sent-date>
    <item-id>4a00000180001114</item-id>
    <workflow-id>4d00000180001114</workflow-id>
    <workflow-task-type>UNKNOWN</workflow-task-type>
    <performer>Administrator</performer>
```

```

<stamp>1b0000018000e550</stamp>
<links>
    <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/REPO/queue-tasks/
        4a00000180001114"></link>
    <link rel="http://identifiers.emc.com/linkrel/workflow"
        href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180001114"></link>
    <link rel="http://identifiers.emc.com/linkrel/workflow-activity"
        href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-activities/4c0000018001f578"></link>
    <link rel="http://identifiers.emc.com/linkrel/inbox-item"
        href="http://localhost:8080/dctm-rest/repositories/REPO/inbox/
        1b0000018000e550"></link>
</links>
</work-queue-task>

```



#### Example 2-419: JSON response

```
{
    "name": "work-queue-task",
    "original-wq-name": "test-work-queue-2",
    "pulled": true,
    "held": false,
    "held-by": "",
    "wq-policy-id": "0000000180000d1d",
    "wq-doc-profile": "",
    "increment-priority": 1,
    "skill-level": 0,
    "process-id": "4b0000018001f56c",
    "is-signoff-required": false,
    "is-repeatable": true,
    "is-delegable": true,
    "is-end-activity": true,
    "is-distribution-listtsak": false,
    "is-group-owned": false,
    "runtime-state": "DORMANT",
    "is-failed-autotask": false,
    "is-manual-transition": false,
    "instruction": "",
    "is-sequential-activity": false,
    "has-self-reject-link": false,
    "is-first-performer": false,
    "is-last-performer": false,
    "has-forward-activities": false,
    "has-reject-activities": false,
    "queue-item-id": "1b0000018000e550",
    "activity-id": "4c0000018001f578",
    "activity-name": "work queue2",
    "is-performer-assignment-required": false,
    "complete-error-status": "COMPLETE_UNKNOWN",
    "task-name": "work queue2",
    "task-type": "WORKFLOWTASK",
    "task-state": "DORMANT",
    "priority": 171,
    "sent-date": "2019-06-06T05:00:29.000+00:00",
    "item-id": "4a00000180001114",
    "workflow-id": "4d00000180001114",
    "workflow-task-type": "UNKNOWN",
    "performer": "Administrator",
    "stamp": "1b0000018000e550",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            queue-tasks/4a00000180001114"
        }
    ]
}
```

```

        "rel": "http://identifiers.emc.com/linkrel/workflow",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180001114"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-activities/4c0000018001f578"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/inbox-item",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/inbox/
        1b0000018000e550"
    }
]
}

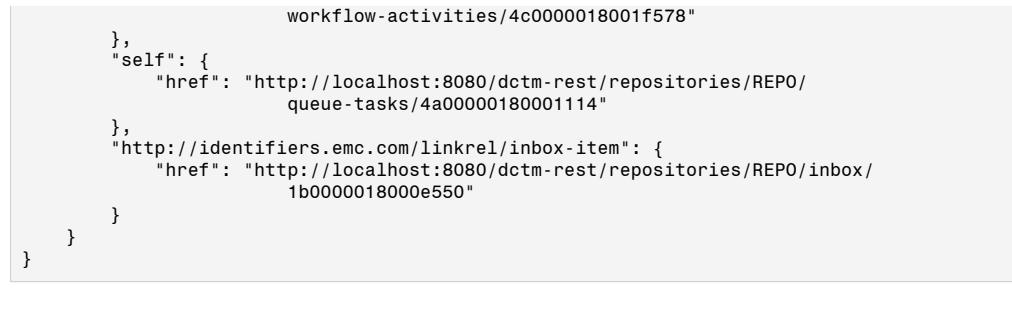
```



#### ➤ Example 2-420: HAL: JSON Response

```
{
    "name": "work-queue-task",
    "original-wq-name": "test-work-queue-2",
    "pulled": true,
    "held": false,
    "held-by": "",
    "wq-policy-id": "0000000180000d1d",
    "wq-doc-profile": "",
    "increment-priority": 1,
    "skill-level": 0,
    "process-id": "4b0000018001f56c",
    "is-signoff-required": false,
    "is-repeatable": true,
    "is-delegable": true,
    "is-end-activity": true,
    "is-distribution-listtsak": false,
    "is-group-owned": false,
    "runtime-state": "DORMANT",
    "is-failed-autotask": false,
    "is-manual-transition": false,
    "instruction": "",
    "is-sequential-a-activity": false,
    "has-self-reject-link": false,
    "is-first-performer": false,
    "is-last-performer": false,
    "has-forward-activities": false,
    "has-reject-activities": false,
    "queue-item-id": "1b0000018000e550",
    "activity-id": "4c0000018001f578",
    "activity-name": "work queue2",
    "is-performer-assignment-required": false,
    "complete-error-status": "COMPLETE_UNKNOWN",
    "task-name": "work queue2",
    "task-type": "WORKFLOWTASK",
    "task-state": "DORMANT",
    "priority": 171,
    "sent-date": "2019-06-06T05:00:29.000+00:00",
    "item-id": "4a00000180001114",
    "workflow-id": "4d00000180001114",
    "workflow-task-type": "UNKNOWN",
    "performer": "Administrator",
    "stamp": "1b0000018000e550",
    "_links": {
        "http://identifiers.emc.com/linkrel/workflow": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d00000180001114"
        },
        "http://identifiers.emc.com/linkrel/workflow-activity": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/

```

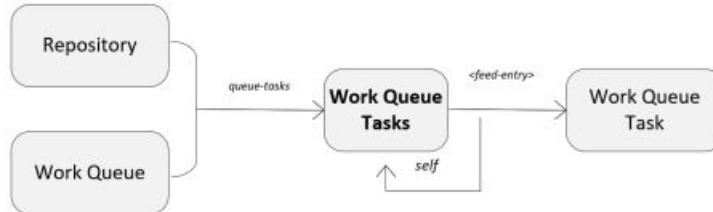


## 2.88.2 Work Queue Tasks

The *Work Queue Tasks* resource is used to get a collection of work queues to which a specific user belongs.

### 2.88.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams](#) on page 13.

### 2.88.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Work queue tasks
- Feed Updated: Server's current time
- Supports POST: No
- Entry: “[Work Queue Task](#)” on page 887 resource
- Entry ID: The entity URI
- Entry Title: The name (*original-wq-name*) of the Work Queue
- Entry Updated: The current time of the server

### 2.88.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	"Work Queues" on page 882

### 2.88.2.4 Operations

#### 2.88.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves Work Queues

#### 2.88.2.4.2 Get work queues

This operation retrieves a list of Work Queue Tasks that belong to the specified user.

##### 2.88.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	inline, links	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			
	user-name	Specifies the name of the user	String	N/A	The user that is currently logged in
	queue-names	Get the queue tasks under the work queues to which the specified user belongs.	String	existing work queues	All work queues to which a specified user belongs.

HTTP method	Query parameters	Description	Data type	Value range	Default value
	filter-type	Specify task type to be returned, valid values are not case-sensitive: <ul style="list-style-type: none"><li>• <b>ALL:</b> Shows all tasks in the work queue.</li><li>• <b>UNASSIGNED:</b> Shows unassigned tasks in the work queue.</li><li>• <b>ELIGIBLE_AND_OWNED:</b> Shows the tasks that the processor can work on and the tasks that the processor owns in the work queue.</li><li>• <b>ELIGIBLE:</b> Shows the tasks that the processor can work on in the work queue.</li></ul>	String	One of the valid values.	ELIGIBLE_AND_OWNERD

#### 2.88.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.88.2.4.2.3 Request media types

N/A

#### 2.88.2.4.2.4 Request body

N/A

#### 2.88.2.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.88.2.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.88.2.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.88.2.4.2.8 Response body

##### Example 2-421: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/
    queue-tasks</id>
  <title>Work queue tasks</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2019-06-13T03:11:13.272+00:00</updated>
  <link rel="self"
        href="http://localhost:8080/dctm-rest/repositories/
          REPO/queue-tasks"></link>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/
      REPO/queue-tasks/4a00000180001112</id>
    <title>test work queue</title>
    <author>
      <name>dmadmin</name>
      <uri>http://localhost:8080/dctm-rest/repositories/
        REPO/users/dmadmin</uri>
    </author>
    <summary>4a00000180001112 170</summary>
    <updated>2019-06-06T04:59:06.000+00:00</updated>
    <published>2019-06-06T04:59:06.000+00:00</published>
    <link rel="edit"
          href="http://localhost:8080/dctm-rest/
            repositories/REPO/queue-tasks/
              4a00000180001112"></link>
    <content type="application/vnd.emc.documentum+xml"
      src="http://localhost:8080/dctm-rest/"/>
```

```
repositories/REPO/queue-tasks/
4a00000180001112"></content>
</entry>
<entry>
<id>http://localhost:8080/dctm-rest/repositories/REPO/
queue-tasks/4a00000180001113</id>
<title>test work queue</title>
<author>
<name>test work queue</name>
<uri>http://localhost:8080/dctm-rest/repositories/
REPO/users/test%2Bwork%2Bqueue</uri>
</author>
<summary>4a00000180001113 170</summary>
<updated>2019-06-06T04:59:17.000+00:00</updated>
<published>2019-06-06T04:59:17.000+00:00</published>
<link rel="edit"
      href="http://localhost:8080/dctm-rest/
      repositories/REPO/queue-tasks/
      4a00000180001113"></link>
<content type="application/vnd.emc.documentum+xml"
         src="http://localhost:8080/dctm-rest/
         repositories/REPO/queue-tasks/
         4a00000180001113"></content>
</entry>
<entry>
<id>http://localhost:8080/dctm-rest/repositories/REPO/
queue-tasks/4a00000180001114</id>
<title>test-work-queue-2</title>
<author>
<name>test-work-queue-2</name>
<uri>http://localhost:8080/dctm-rest/repositories/
REPO/users/test-work-queue-2</uri>
</author>
<summary>4a00000180001114 170</summary>
<updated>2019-06-06T05:00:29.000+00:00</updated>
<published>2019-06-06T05:00:29.000+00:00</published>
<link rel="edit"
      href="http://localhost:8080/dctm-rest/
      repositories/REPO/queue-tasks/
      4a00000180001114"></link>
<content type="application/vnd.emc.documentum+xml"
         src="http://localhost:8080/dctm-rest/
         repositories/REPO/queue-tasks/
         4a00000180001114"></content>
</entry>
<entry>
<id>http://localhost:8080/dctm-rest/repositories/REPO/
queue-tasks/4a00000180001115</id>
<title>test-work-queue-2</title>
<author>
<name>test-work-queue-2</name>
<uri>http://localhost:8080/dctm-rest/repositories/
REPO/users/test-work-queue-2</uri>
</author>
<summary>4a00000180001115 169</summary>
<updated>2019-06-06T05:52:43.000+00:00</updated>
<published>2019-06-06T05:52:43.000+00:00</published>
<link rel="edit"
      href="http://localhost:8080/dctm-rest/
      repositories/REPO/queue-tasks/
      4a00000180001115"></link>
<content type="application/vnd.emc.documentum+xml"
         src="http://localhost:8080/dctm-rest/
         repositories/REPO/queue-tasks/
         4a00000180001115"></content>
</entry>
</feed>
```



▶ **Example 2-422: JSON response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/
    REPO/queue-tasks",
  "title": "Work queue tasks",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-06-13T03:13:37.441+00:00",
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/
        repositories/REPO/queue-tasks"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/
        repositories/REPO/queue-tasks/
        4a00000180001112",
      "title": "test work queue",
      "author": [
        {
          "name": "dmadmin",
          "uri": "http://localhost:8080/dctm-rest/
            repositories/REPO/users/dmadmin"
        }
      ],
      "summary": "4a00000180001112 170",
      "updated": "2019-06-06T04:59:06.000+00:00",
      "published": "2019-06-06T04:59:06.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/
            repositories/REPO/queue-tasks/
            4a00000180001112"
        }
      ],
      "content": {
        "type": "application/json",
        "src": "http://localhost:8080/dctm-rest/
          repositories/REPO/queue-tasks/
          4a00000180001112"
      }
    },
    {
      "id": "http://localhost:8080/dctm-rest/
        repositories/REPO/queue-tasks/
        4a0000018000110c",
      "title": "test-work-queue-2",
      "author": [
        {
          "name": "sysadmin",
          "uri": "http://localhost:8080/dctm-rest/
            repositories/REPO/users/
            sysadmin"
        }
      ],
      "summary": "4a0000018000110c 196",
      "updated": "2019-06-05T03:05:10.000+00:00",
      "published": "2019-06-05T03:05:10.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/
            repositories/REPO/queue-tasks/
            4a0000018000110c"
        }
      ],
      "content": {
        "type": "application/json",
        "src": "http://localhost:8080/dctm-rest/"
      }
    }
  ]
}
```

```
        repositories/REPO/queue-tasks/
        4a0000018000110c"
    },
    {
        "id": "http://localhost:8080/dctm-rest/
            repositories/REPO/queue-tasks/
            4a00000180001114",
        "title": "test-work-queue-2",
        "author": [
            {
                "name": "test-work-queue-2",
                "uri": "http://localhost:8080/dctm-rest/
                    repositories/REPO/users/
                    test-work-queue-2"
            }
        ],
        "summary": "4a00000180001114 170",
        "updated": "2019-06-06T05:00:29.000+00:00",
        "published": "2019-06-06T05:00:29.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/queue-tasks/
                    4a00000180001114"
            }
        ],
        "content": {
            "type": "application/json",
            "src": "http://localhost:8080/dctm-rest/
                repositories/REPO/queue-tasks/
                4a00000180001114"
        }
    },
    {
        "id": "http://localhost:8080/dctm-rest/
            repositories/REPO/queue-tasks/
            4a00000180001115",
        "title": "test-work-queue-2",
        "author": [
            {
                "name": "test-work-queue-2",
                "uri": "http://localhost:8080/dctm-rest/
                    repositories/REPO/users/
                    test-work-queue-2"
            }
        ],
        "summary": "4a00000180001115 169",
        "updated": "2019-06-06T05:52:43.000+00:00",
        "published": "2019-06-06T05:52:43.000+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "http://localhost:8080/dctm-rest/
                    repositories/REPO/queue-tasks/
                    4a00000180001115"
            }
        ],
        "content": {
            "type": "application/json",
            "src": "http://localhost:8080/dctm-rest/
                repositories/REPO/queue-tasks/
                4a00000180001115"
        }
    }
]
```



▶ **Example 2-423: HAL: JSON Response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/queue-tasks",
  "title": "Work queue tasks",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-06-13T03:15:27.348+00:00",
  "_embedded": {
    "collection": [
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                     queue-tasks/4a00000180001112",
        "title": "test work queue",
        "author": [
          {
            "name": "dmadmin",
            "uri": "http://localhost:8080/dctm-rest/repositories/
                   REPO/users/dmadmin"
          }
        ],
        "summary": "4a00000180001112 170",
        "updated": "2019-06-06T04:59:06.000+00:00",
        "published": "2019-06-06T04:59:06.000+00:00",
        "_links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
                   REPO/queue-tasks/4a00000180001112"
          }
        }
      },
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                     queue-tasks/4a0000018000110c",
        "title": "test-work-queue-2",
        "author": [
          {
            "name": "sysadmin",
            "uri": "http://localhost:8080/dctm-rest/repositories/
                   REPO/users/sysadmin"
          }
        ],
        "summary": "4a0000018000110c 196",
        "updated": "2019-06-05T03:05:10.000+00:00",
        "published": "2019-06-05T03:05:10.000+00:00",
        "_links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
                   REPO/queue-tasks/4a0000018000110c"
          }
        }
      },
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                     queue-tasks/4a00000180001114",
        "title": "test-work-queue-2",
        "author": [
          {
            "name": "test-work-queue-2",
            "uri": "http://localhost:8080/dctm-rest/repositories/
                   REPO/users/test-work-queue-2"
          }
        ],
        "summary": "4a00000180001114 170",
        "updated": "2019-06-06T05:00:29.000+00:00",
        "published": "2019-06-06T05:00:29.000+00:00",
        "_links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/
                   REPO/queue-tasks/4a00000180001114"
          }
        }
      }
    ]
  }
}
```

```

        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
            queue-tasks/4a00000180001115",
        "title": "test-work-queue-2",
        "author": [
            {
                "name": "test-work-queue-2",
                "uri": "http://localhost:8080/dctm-rest/repositories/
                    REPO/users/test-work-queue-2"
            }
        ],
        "summary": "4a00000180001115 169",
        "updated": "2019-06-06T05:52:43.000+00:00",
        "published": "2019-06-06T05:52:43.000+00:00",
        "_links": {
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/
                    REPO/queue-tasks/4a00000180001115"
            }
        }
    ]
},
"_links": {
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            queue-tasks"
    }
}
}

```



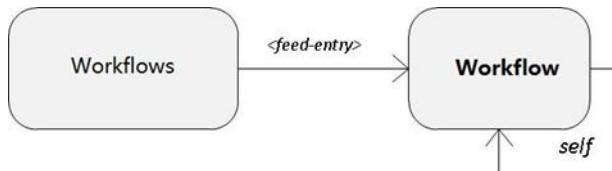
## 2.89 Workflow(s)

### 2.89.1 Workflow

The *Workflow* resource is used to get a specified Workflow instance.

#### 2.89.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Workflows](#)” on page 920 and [About the Diagrams](#) on page 13.

### 2.89.1.2 Feed

No feed support.

### 2.89.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">"Workflow" on page 906</a>
workflow-template [1]	A workflow template	<a href="#">"Workflow Template" on page 959</a>
workflow-activity [1]	The workflow activity	<a href="#">"Workflow Activity" on page 935</a>
workflow-attachments [1]	The workflow attachments	<a href="#">"Workflow Attachments" on page 952</a>
work-items [1]	The work items	<a href="#">"Work Items" on page 874</a>
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

### 2.89.1.4 Operations

#### 2.89.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a Workflow instance
PUT	Performs Workflow actions

#### 2.89.1.4.2 Get a Workflow instance

This operation retrieves an instance of a Workflow.

##### 2.89.1.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### **2.89.1.4.2.2 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.89.1.4.2.3 Request media types**

N/A

#### **2.89.1.4.2.4 Request body**

N/A

#### **2.89.1.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.89.1.4.2.6 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.89.1.4.2.7 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.89.1.4.2.8 Response body

➡ Example 2-424: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<workflow
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_workflow">
<properties>
    <object_name>a_test_workflow 2019-02-26</object_name>
    <process_id>4b0000018000236e</process_id>
    <r_creator_name>dmadmin</r_creator_name>
    <supervisor_name>dmadmin</supervisor_name>
    <r_runtime_state>1</r_runtime_state>
    <r_start_date>2019-02-26T06:35:58.000+00:00</r_start_date>
    <i_next_seqno>1</i_next_seqno>
    <r_alias_set_id>0000000000000000</r_alias_set_id>
    <parent_id>0000000000000000</parent_id>
    <parent_act_seqno>0</parent_act_seqno>
    <correlation_identifier>
        9f000661a430d930db3138306730303f
    </correlation_identifier>
    <i_next_act_group_seqno>0</i_next_act_group_seqno>
    <r_act_seqno>
        <item>0</item>
    </r_act_seqno>
    <r_act_name>
        <item>Activity-1</item>
    </r_act_name>
    <r_act_def_id>
        <item>4c000001800025c9</item>
    </r_act_def_id>
    <r_act_errno>
        <item>0</item>
    </r_act_errno>
    <r_act_state>
        <item>1</item>
    </r_act_state>
    <r_repeat_invoke>
        <item>true</item>
    </r_repeat_invoke>
    <r_trigger_thresh>
        <item>0</item>
    </r_trigger_thresh>
    <r_trigger_input>
        <item>0</item>
    </r_trigger_input>
    <r_trigger_revert>
        <item>0</item>
    </r_trigger_revert>
    <r_pre_timer>
        <item xsi:nil="true"/>
    </r_pre_timer>
    <r_post_timer>
        <item xsi:nil="true"/>
    </r_post_timer>
    <r_total_witem>
        <item>1</item>
    </r_total_witem>
    <r_complete_witem>
        <item></item>
    </r_complete_witem>
    <r_last_performer>
        <item/>
    </r_last_performer>
    <r_last_witem_id>
        <item>0000000000000000</item>
    </r_last_witem_id>

```

```

</r_last_witem_id>
<i_performer_flag>
    <item>0</item>
</i_performer_flag>
<r_target_act_seqno>
    <item>-1</item>
</r_target_act_seqno>
<i_is_replica>false</i_is_replica>
<i_vstamp>1</i_vstamp>
<r_object_id>4d0000018000010a</r_object_id>
</properties>
<links>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a"
        rel="self"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e"
        rel="http://identifiers.emc.com/linkrel/workflow-template"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments"
        rel="http://identifiers.emc.com/linkrel/workflow-attachments"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/work-items"
        rel="http://identifiers.emc.com/linkrel/work-items"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-activities/4c000001800025c9"
        rel="http://identifiers.emc.com/linkrel/workflow-activity"
        title="Activity-1"/>
</links>
</workflow>

```



### Example 2-425: JSON response

```
{
    "name": "workflow",
    "type": "dm_workflow",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow",
    "properties": {
        "object_name": "a_test_workflow 2019-02-26",
        "process_id": "4b0000018000236e",
        "r_creator_name": "dmadmin",
        "supervisor_name": "dmadmin",
        "r_runtime_state": 1,
        "r_start_date": "2019-02-26T06:35:58.000+00:00",
        "i_next_seqno": 1,
        "r_alias_set_id": "0000000000000000",
        "parent_id": "0000000000000000",
        "parent_act_seqno": 0,
        "correlation_identifier": "9f000661a430d930db3138306730303f",
        "i_next_act_group_seqno": 0,
        "r_act_seqno": [
            0
        ],
        "r_act_name": [
            "Activity-1"
        ],
        "r_act_def_id": [
            "4c000001800025c9"
        ],
        "r_act_errno": [
            0
        ],
        "r_act_state": [
            1
        ],
    }
}
```

```

    "r_repeat_invoke": [
        true
    ],
    "r_trigger_thresh": [
        0
    ],
    "r_trigger_input": [
        0
    ],
    "r_trigger_revert": [
        0
    ],
    "r_pre_timer": [
        null
    ],
    "r_post_timer": [
        null
    ],
    "r_total_witem": [
        1
    ],
    "r_complete_witem": [
        0
    ],
    "r_last_performer": [
        ""
    ],
    "r_last_witem_id": [
        "0000000000000000"
    ],
    "i_performer_flag": [
        0
    ],
    "r_target_act_seqno": [
        -1
    ],
    "i_is_replica": false,
    "i_vstamp": 1,
    "r_object_id": "4d0000018000010a"
},
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-template",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-attachments",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/work-items",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/work-items"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-1",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-activities/4c000001800025c9"
    }
]
}

```



 Example 2-426: HAL: JSON Response

```
{  
    "name": "workflow",  
    "type": "dm_workflow",  
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/  
                  dm_workflow",  
    "properties": {  
        "object_name": "a_test_workflow 2019-02-26",  
        "process_id": "4b0000018000236e",  
        "r_creator_name": "dmadmin",  
        "supervisor_name": "dmadmin",  
        "r_runtime_state": 1,  
        "r_start_date": "2019-02-26T06:35:58.000+00:00",  
        "i_next_seqno": 1,  
        "r_alias_set_id": "0000000000000000",  
        "parent_id": "0000000000000000",  
        "parent_act_seqno": 0,  
        "correlation_identifier": "9f000661a430d930db3138306730303f",  
        "i_next_act_group_seqno": 0,  
        "r_act_seqno": [  
            0  
        ],  
        "r_act_name": [  
            "Activity-1"  
        ],  
        "r_act_def_id": [  
            "4c000001800025c9"  
        ],  
        "r_act_errno": [  
            0  
        ],  
        "r_act_state": [  
            1  
        ],  
        "r_repeat_invoke": [  
            true  
        ],  
        "r_trigger_thresh": [  
            0  
        ],  
        "r_trigger_input": [  
            0  
        ],  
        "r_trigger_revert": [  
            0  
        ],  
        "r_pre_timer": [  
            null  
        ],  
        "r_post_timer": [  
            null  
        ],  
        "r_total_witem": [  
            1  
        ],  
        "r_complete_witem": [  
            0  
        ],  
        "r_last_performer": [  
            ""  
        ],  
        "r_last_witem_id": [  
            "0000000000000000"  
        ],  
        "i_performer_flag": [  
            0  
        ],  
        "r_target_act_seqno": [  
    }  
}
```

```
-1
],
  "i_is_replica": false,
  "i_vstamp": 1,
  "r_object_id": "4d0000018000010a"
},
"_links": {
  "http://identifiers.emc.com/linkrel/work-items": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
4d0000018000010a/work-items"
  },
  "http://identifiers.emc.com/linkrel/workflow-activity": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
workflow-activities/4c000001800025c9",
    "title": "Activity-1"
  },
  "self": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
workflows/4d0000018000010a"
  },
  "http://identifiers.emc.com/linkrel/workflow-attachments": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
4d0000018000010a/attachments"
  },
  "http://identifiers.emc.com/linkrel/workflow-template": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
workflow-templates/4b0000018000236e"
  }
}
```



#### **2.89.1.4.3 Perform actions on a Workflow instance**

This operation performs actions on a Workflow instance.

#### **2.89.1.4.3.1 HTTP Request Method and Query Parameters**

HTTP method	Query parameters	Description	Data type	Value range	Default value
PUT	action	<p>The specified action that is performed on the Workflow instance. The following actions are supported:</p> <ul style="list-style-type: none"> <li>• HALT</li> <li>• HALT-ALL</li> <li>• RESUME</li> <li>• RESUME-ALL</li> <li>• ABORT</li> <li>• CHANGE - SUPERVISOR</li> </ul> <p>This parameter is required.</p>	string	One of the supported actions	null

HTTP method	Query parameters	Description	Data type	Value range	Default value
	activity	The name of the activity to be halted, resumed, or restarted.  This parameter is required when the action is one of: <i>HALT</i> , <i>RESUME</i>	string	N/A	null
	supervisor	The name of the new supervisor.  This parameter is required when the action specified is <i>CHANGE-SUPERVISOR</i>	string	N/A	null

#### 2.89.1.4.3.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.89.1.4.3.3 Request media types

N/A

#### 2.89.1.4.3.4 Request body

N/A

#### 2.89.1.4.3.5 Response status

- 200 - Action performed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.89.1.4.3.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.89.1.4.3.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.89.1.4.3.8 Response body

The response body is the same as that of the GET operation. We have included here for your convenience.

##### Example 2-427: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_workflow">
<properties>
    <object_name>a_test_workflow_2019-02-26</object_name>
    <process_id>4b0000018000236e</process_id>
    <r_creator_name>dmadmin</r_creator_name>
    <supervisor_name>dmadmin</supervisor_name>
    <r_runtime_state>1</r_runtime_state>
    <r_start_date>2019-02-26T06:35:58.000+00:00</r_start_date>
    <i_next_seqno>1</i_next_seqno>
    <r_alias_set_id>0000000000000000</r_alias_set_id>
    <parent_id>0000000000000000</parent_id>
    <parent_act_seqno>0</parent_act_seqno>
    <correlation_identifier>
        9f000661a430d930db3138306730303f
    </correlation_identifier>
    <i_next_act_group_seqno>0</i_next_act_group_seqno>
    <r_act_seqno>
        <item>0</item>
    </r_act_seqno>
    <r_act_name>
        <item>Activity-1</item>
    </r_act_name>
    <r_act_def_id>
        <item>4c000001800025c9</item>
    </r_act_def_id>
    <r_act_errno>
        <item>0</item>
    </r_act_errno>
    <r_act_state>
        <item>1</item>
    </r_act_state>
    <r_repeat_invoke>
        <item>true</item>
    </r_repeat_invoke>

```

```

</r_repeat_invoke>
<r_trigger_thresh>
    <item>0</item>
</r_trigger_thresh>
<r_trigger_input>
    <item>0</item>
</r_trigger_input>
<r_trigger_revert>
    <item>0</item>
</r_trigger_revert>
<r_pre_timer>
    <item xsi:nil="true"/>
</r_pre_timer>
<r_post_timer>
    <item xsi:nil="true"/>
</r_post_timer>
<r_total_witem>
    <item>1</item>
</r_total_witem>
<r_complete_witem>
    <item>0</item>
</r_complete_witem>
<r_last_performer>
    <item/>
</r_last_performer>
<r_last_witem_id>
    <item>0000000000000000</item>
</r_last_witem_id>
<i_performer_flag>
    <item>0</item>
</i_performer_flag>
<r_target_act_seqno>
    <item>-1</item>
</r_target_act_seqno>
<i_is_replica>false</i_is_replica>
<i_vstamp>1</i_vstamp>
<r_object_id>4d0000018000010a</r_object_id>
</properties>
<links>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a"
        rel="self"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e"
        rel="http://identifiers.emc.com/linkrel/workflow-template"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments"
        rel="http://identifiers.emc.com/linkrel/workflow-attachments"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/work-items"
        rel="http://identifiers.emc.com/linkrel/work-items"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-activities/4c000001800025c9"
        rel="http://identifiers.emc.com/linkrel/workflow-activity"
        title="Activity-1"/>
</links>
</workflow>

```



### Example 2-428: JSON response

```
{
    "name": "workflow",
    "type": "dm_workflow",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow",
    "properties": {

```

```

"object_name": "a_test_workflow_2019-02-26",
"process_id": "4b0000018000236e",
"r_creator_name": "dmadmin",
"supervisor_name": "dmadmin",
"r_runtime_state": 1,
"r_start_date": "2019-02-26T06:35:58.000+00:00",
"i_next_seqno": 1,
"r_alias_set_id": "0000000000000000",
"parent_id": "0000000000000000",
"parent_act_seqno": 0,
"correlation_identifier": "9f000661a430d930db3138306730303f",
"i_next_act_group_seqno": 0,
"r_act_seqno": [
    0
],
"r_act_name": [
    "Activity-1"
],
"r_act_def_id": [
    "4c000001800025c9"
],
"r_act_errno": [
    0
],
"r_act_state": [
    1
],
"r_repeat_invoke": [
    true
],
"r_trigger_thresh": [
    0
],
"r_trigger_input": [
    0
],
"r_trigger_revert": [
    0
],
"r_pre_timer": [
    null
],
"r_post_timer": [
    null
],
"r_total_witem": [
    1
],
"r_complete_witem": [
    0
],
"r_last_performer": [
    ""
],
"r_last_witem_id": [
    "0000000000000000"
],
"i_performer_flag": [
    0
],
"r_target_act_seqno": [
    -1
],
"i_is_replica": false,
"i_vstamp": 1,
"r_object_id": "4d0000018000010a"
},
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/"
    }
]

```

```

        "4d0000018000010a"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-template",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-attachments",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d0000018000010a/attachments"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/work-items",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d0000018000010a/work-items"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-1",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9"
    }
]
}

```



### Example 2-429: HAL: JSON Response

```
{
    "name": "workflow",
    "type": "dm_workflow",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow",
    "properties": {
        "object_name": "a_test_workflow 2019-02-26",
        "process_id": "4b0000018000236e",
        "r_creator_name": "dmadmin",
        "supervisor_name": "dmadmin",
        "r_runtime_state": 1,
        "r_start_date": "2019-02-26T06:35:58.000+00:00",
        "i_next_seqno": 1,
        "r_alias_set_id": "0000000000000000",
        "parent_id": "0000000000000000",
        "parent_act_seqno": 0,
        "correlation_identifier": "9f000661a430d930db3138306730303f",
        "i_next_act_group_seqno": 0,
        "r_act_seqno": [
            0
        ],
        "r_act_name": [
            "Activity-1"
        ],
        "r_act_def_id": [
            "4c000001800025c9"
        ],
        "r_act_errno": [
            0
        ],
        "r_act_state": [
            1
        ],
        "r_repeat_invoke": [
            true
        ],
        "r_trigger_thresh": [
            0
        ],
        "r_trigger_input": [
            0
        ],
        "r_trigger_output": [
            0
        ]
    }
}
```

```

        "r_trigger_revert": [
            0
        ],
        "r_pre_timer": [
            null
        ],
        "r_post_timer": [
            null
        ],
        "r_total_witem": [
            1
        ],
        "r_complete_witem": [
            0
        ],
        "r_last_performer": [
            ""
        ],
        "r_last_witem_id": [
            "0000000000000000"
        ],
        "i_performer_flag": [
            0
        ],
        "r_target_act_seqno": [
            -1
        ],
        "i_is_replica": false,
        "i_vstamp": 1,
        "r_object_id": "4d0000018000010a"
    },
    "_links": {
        "http://identifiers.emc.com/linkrel/work-items": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/4d0000018000010a/work-items"
        },
        "http://identifiers.emc.com/linkrel/workflow-activity": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-activities/4c000001800025c9",
            "title": "Activity-1"
        },
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/4d0000018000010a"
        },
        "http://identifiers.emc.com/linkrel/workflow-attachments": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/4d0000018000010a/attachments"
        },
        "http://identifiers.emc.com/linkrel/workflow-template": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates/4b0000018000236e"
        }
    }
}

```

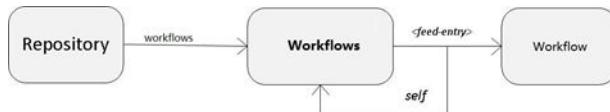


## 2.89.2 Workflows

The *Workflows* resource is used to get *Workflow* instances as a feed.

### 2.89.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams](#) on page 13.

### 2.89.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Workflows
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: Workflow resource
- Entry ID: The entry URI
- Entry Title: The name of the Workflow
- Entry Updated: The last modify date

### 2.89.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">“Workflows” on page 920</a>

### 2.89.2.4 Operations

#### 2.89.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves Workflows as a feed
POST	Used to start a Workflow

#### 2.89.2.4.2 Get Workflows

This operation retrieves Workflows as a feed.

##### 2.89.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, include-total, links, sort, view, filter, inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.89.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.89.2.4.2.3 Request media types

N/A

##### 2.89.2.4.2.4 Request body

N/A

##### 2.89.2.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.89.2.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.89.2.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.89.2.4.2.8 Response body

##### ➡ Example 2-430: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/workflows</id>
    <title>Workflows</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2019-02-26T06:20:35.128+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
        100
    </dm:items-per-page>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/workflows"
        rel="self"/>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d00000180000100</id>
        <title>a_test_workflow 2018-12-25</title>
        <updated>2019-02-26T06:20:35.146+00:00</updated>
        <published>2018-12-25T09:18:22.000+00:00</published>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d00000180000100"
            rel="edit"/>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-templates/4b0000018000236e"
            rel="http://identifiers.emc.com/linkrel/workflow-template"/>
        <content
            src="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d00000180000100"
            type="application/vnd.emc.documentum+xml"/>
    </entry>
</feed>
```



▶ **Example 2-431: JSON response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows",
  "title": "Workflows",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-26T06:23:05.510+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d00000180000100",
      "title": "a_test_workflow 2018-12-25",
      "updated": "2019-02-26T06:23:05.510+00:00",
      "published": "2018-12-25T09:18:22.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflows/4d00000180000100"
        },
        {
          "rel": "http://identifiers.emc.com/linkrel/workflow-template",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
          workflows/4d00000180000100"
      }
    }
  ]
}
```



▶ **Example 2-432: HAL: JSON Response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows",
  "title": "Workflows",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-26T06:23:21.597+00:00",
  "page": 1,
  "items-per-page": 100,
  "_embedded": {
    "collection": [
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
          4d00000180000100",
        "title": "a_test_workflow 2018-12-25",
        "updated": "2019-02-26T06:23:21.597+00:00",
        "published": "2018-12-25T09:18:22.000+00:00",
        "links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              workflows/4d00000180000100"
          }
        }
      }
    ]
  }
}
```

```

        },
        "http://identifiers.emc.com/linkrel/workflow-template": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-templates/4b0000018000236e"
        }
    }
}
],
"_links": {
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows"
    }
}
}

```



#### 2.89.2.4.3 Start Workflows

This operation creates and then starts a workflow from a template or it can be used to create and then start a quick flow that is not based on a workflow template. Quick flows can be used to route objects to users or groups.

##### 2.89.2.4.3.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST	quick-flow	<p>Specifies whether to start an ad-hoc workflow or start a workflow from a template.</p> <ul style="list-style-type: none"> <li><i>true</i>: Creates and then starts an ad-hoc workflow to route objects to users and groups</li> <li><i>false</i>: Creates and then starts a workflow from a template</li> </ul>	string	<i>true, false</i>	<i>false</i>

##### 2.89.2.4.3.2 Request headers

- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.89.2.4.3.3 Request media types

- application/vnd.emc.documentum+xml
- application/vnd.emc.documentum+json
- application/hal+json

#### 2.89.2.4.3.4 Request body

A Workflow can be created and started ad-hoc or using a template.

##### Start a Workflow from a template

The default behavior is to create and start a Workflow from a template. In this case, creating a reference to the Workflow template in XML or JSON is mandatory. You can use the *workflow-name* property to specify the object name of the Workflow instance.

When the Workflow is created with a start activity that does not have any input ports, the start activity becomes active and then appears in the recipient's inbox.

When the Workflow is designed with a start activity that does contain an input port, the start activity does not become active until you attach a package to its input port. To do this, just add a package to a start activity during the workflow instance creation from a template. When an activity name is not provided, a default start activity is used.

To add documents or notes to the packages in the template definition, with the exception of the mandatory *href* attribute, you can use a list of packages, and one of those packages can contain the following attributes:

- **package-name:** The name of a workflow package. This attribute is mandatory.
- **document-ids:** The object IDs of the package's components. This attribute is only mandatory when adding a package action.
- **note#text:** Defines a message to send to the package's recipients.
- **note#persistent:** Set this to *TRUE* when you want to send the note defined in *note#text* to the performers of all activities receiving this package. When this is set to *FALSE*, the note is sent only to the performers of the activities that immediately follow the start activity.

##### Starts an ad-hoc Workflow

When you are starting an ad-hoc flow, the *quick-flow* Request parameter must be set to *true*. This method allows you to create and start a workflow that is not based on a workflow template. Creating and starting a workflow this way enables you to route one or more objects to one or more users, groups, or users and groups.

When you send tasks to multiple users, you can select whether each user receives those tasks simultaneously or sequentially.

The following parameters can be set when starting an ad-hoc workflow:

- **users:** A list of users who are assigned workflow tasks
- **groups:** A list of groups that are assigned workflow tasks
- **object-ids:** This mandatory attribute is a list of documents that you want to send to the specified users and groups. You can only send persistent objects in a workflow.
- **instruction:** The task instructions for the recipients of workflow tasks.
- **priority:** A representation of the urgency or importance of a task. The values that are valid as priority levels include: *LOW*, *MEDIUM*, and *HIGH*. Each value is case sensitive.
- **flags:** Any combination of the options shown in the table below, bitwise OR'ed together. For example 1, 2, 3, 4, 7, 12, 20...

Name in IDfWorkflow (Foundation Java API interface)	Description	Value
<i>DF_REQ_END_NOTIFICATION</i>	Use this flag when you want the sender to receive a notification when all recipients have completed their workflow tasks	0x0001
<i>DF_REQ_SIGN_OFF</i>	Use this flag when you want the recipients to sign off the workflow tasks	0x0002
<i>DF_SEQUENTIAL</i>	Use this flag when you want the recipients to receive the workflow tasks sequentially	0x0004
<i>DF_ALLOW_REJECT_PREVIOUS</i>	Use this flag when you want the recipients to be able to reject their workflow tasks to the previous recipient if they are in a sequential ad-hoc workflow	0x000c
<i>DF_ALLOW_REJECT_INITIATOR</i>	Use this flag when you want the recipients to be able to reject their workflow tasks to the sender if they are in a sequential ad-hoc workflow	0x0014

### ➡ Example 2-433: XML: Workflow from a template

This creates and starts a workflow from a template, but does not add a package to the start activity.

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow>
    <href>http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates/
        4b0000018000236e</href>
    <workflow-name>the name of this workflow</workflow-name>
</workflow>
```



▶ **Example 2-434: XML: Workflow from a template with a package**

This creates a workflow instance from a template, adds a package to the start activity, and starts the workflow.

```
<workflow>
    <href>http://localhost:8080/dcm-rest/repositories/REPO/workflow-templates/
        4b0000018003e995</href>
    <workflow-name>the name of this workflow</workflow-name>
    <packages>
        <package>
            <package-name>Package0</package-name>
            <document-ids>
                <document-id>0900000180011f0d</document-id>
            </document-ids>
            <notes>
                <note>
                    <text>this is the first note for package0. true-persistent</text>
                    <persistent>true</persistent>
                </note>
                <note>
                    <text>this is the second note for package0. false-persistent</text>
                    <persistent>false</persistent>
                </note>
            </notes>
        </package>
        <package>
            <package-name>Package1</package-name>
            <document-ids>
                <document-id>0900000180011fad</document-id>
            </document-ids>
            <notes>
                <note>
                    <text>this is the first note for package1. true-persistent</text>
                    <persistent>true</persistent>
                </note>
                <note>
                    <text>this is the second note for package1. false-persistent</text>
                    <persistent>false</persistent>
                </note>
            </notes>
        </package>
    </packages>
</workflow>
```



▶ **Example 2-435: XML: Start an ad-hoc workflow**

This creates and then starts an ad-hoc workflow.

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow>
    <users>
        <user>dave</user>
        <user>dmadmin</user>
    </users>
    <groups>
        <group>docu</group>
    </groups>
    <object-ids>
        <object-id>09000001800021bf</object-id>
        <object-id>09000001800021be</object-id>
    </object-ids>
    <instruction>Create and start a quick flow that is not based on a self
        workflow template.</instruction>
    <priority>MEDIUM</priority>
</workflow>
```

```
<flags>3</flags>
</workflow>
```



#### ➤ Example 2-436: JSON: A workflow from a template

This creates and then starts a workflow from a template, but it does not add a package to the start activity.

```
{
  "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates/
  4b0000018000236e",
  "workflow-name": "the name of this workflow"
}
```



#### ➤ Example 2-437: JSON: A workflow from template with package

```
{
  "href": "/repositories/REPO/workflow-templates/4b0000018003e395",
  "packages": [
    {
      "package-name": "Package0",
      "document-ids": [
        "0900000180011f0d"
      ],
      "notes": [
        {
          "text": "this is the first note for package0. true-persistent",
          "persistent": true
        },
        {
          "text": "this is the second note for package0. false-persistent",
          "persistent": false
        }
      ]
    },
    {
      "package-name": "Package1",
      "document-ids": [
        "0900000180011fad"
      ],
      "notes": [
        {
          "text": "this is the first note for package1. true-persistent",
          "persistent": true
        },
        {
          "text": "this is the second note for package1. false-persistent",
          "persistent": false
        }
      ]
    }
  ],
  "workflow-name": "Document review Workflow Instance 2019-7-03"
}
```



#### ➤ Example 2-438: JSON: Start an ad-hoc workflow

This creates and then starts an ad-hoc workflow.

```
{
  "users": [
    "dave",
    "dmadmin",
    "Administrator"
```

```

    ],
    "groups": [
        "docu"
    ],
    "object-ids": [
        "09000001800021c0",
        "09000001800021bf",
        "09000001800021be"
    ],
    "instruction": "A quickflow is a single task you send to one or more users usually for review or acknowledgement use cases. If you send a quickflow to multiple users, you can select whether each user receives the task simultaneously or sequentially.",
    "priority": "LOW",
    "flags": 7
}

```



#### Example 2-439: HAL+JSON: A workflow from a template

This creates and then starts a workflow from a template, but does not add a package to the start activity.

```
{
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e",
    "workflow-name": "the name of this workflow"
}
```



#### Example 2-440: HAL+JSON: A workflow from a template with a package

This creates and then starts a workflow from a template, and adds a package to the start activity of the workflow instance.

```
{
    "href": "/repositories/REPO/workflow-templates/4b0000018003e395",
    "packages": [
        {
            "package-name": "Package0",
            "document-ids": [
                "0900000180011f0d"
            ]
        },
        {
            "notes": [
                {
                    "text": "this is the first note for package0. true-persistent",
                    "persistent": true
                },
                {
                    "text": "this is the second note for package0. false-persistent",
                    "persistent": false
                }
            ]
        },
        {
            "package-name": "Package1",
            "document-ids": [
                "0900000180011fad"
            ],
            "notes": [
                {
                    "text": "this is the first note for package1. true-persistent",
                    "persistent": true
                },
                {
                    "text": "this is the second note for package1. false-persistent",
                    "persistent": false
                }
            ]
        }
    ]
}
```

```
        },
    ],
    "workflow-name": "Document review Workflow Instance 2019-7-03"
}
```



#### ➤ Example 2-441: HAL+JSON: start an ad-hoc workflow

This creates and then starts an ad-hoc workflow.

```
{
  "workflow-name": "the name of this quick flow",
  "users": [
    "dave",
    "dmadmin",
    "Administrator"
  ],
  "groups": [
    "docu"
  ],
  "object-ids": [
    "09000001800021c0",
    "09000001800021bf",
    "09000001800021be"
  ],
  "instruction": "A quickflow is a single task you send to one or more users usually for review or acknowledgement use cases. If you send a quickflow to multiple users, you can select whether each user receives the task simultaneously or sequentially.",
  "priority": "LOW",
  "flags": 7
}
```



##### 2.89.2.4.3.5 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

##### 2.89.2.4.3.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.89.2.4.3.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.89.2.4.3.8 Response body

 **Example 2-442: XML response**

```

<?xml version="1.0" encoding="UTF-8"?>
<workflow
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_workflow">
<properties>
    <object_name>a_test_workflow 2019-02-26</object_name>
    <process_id>4b0000018000236e</process_id>
    <r_creator_name>dmadmin</r_creator_name>
    <supervisor_name>dmadmin</supervisor_name>
    <r_runtime_state>1</r_runtime_state>
    <r_start_date>2019-02-26T06:35:58.000+00:00</r_start_date>
    <i_next_seqno></i_next_seqno>
    <r_alias_set_id>0000000000000000</r_alias_set_id>
    <parent_id>0000000000000000</parent_id>
    <parent_act_seqno>0</parent_act_seqno>
    <correlation_identifier>
        9f000661a430d930db3138306730303f
    </correlation_identifier>
    <i_next_act_group_seqno>0</i_next_act_group_seqno>
    <r_act_seqno>
        <item>0</item>
    </r_act_seqno>
    <r_act_name>
        <item>Activity-1</item>
    </r_act_name>
    <r_act_def_id>
        <item>4c000001800025c9</item>
    </r_act_def_id>
    <r_act_errno>
        <item>0</item>
    </r_act_errno>
    <r_act_state>
        <item>1</item>
    </r_act_state>
    <r_repeat_invoke>
        <item>true</item>
    </r_repeat_invoke>
    <r_trigger_thresh>
        <item>0</item>
    </r_trigger_thresh>
    <r_trigger_input>
        <item>0</item>
    </r_trigger_input>
    <r_trigger_revert>
        <item>0</item>
    </r_trigger_revert>
    <r_pre_timer>
        <item xsi:nil="true" />
    </r_pre_timer>

```

```

<r_post_timer>
    <item xsi:nil="true"/>
</r_post_timer>
<r_total_witem>
    <item>1</item>
</r_total_witem>
<r_complete_witem>
    <item>0</item>
</r_complete_witem>
<r_last_performer>
    <item/>
</r_last_performer>
<r_last_witem_id>
    <item>0000000000000000</item>
</r_last_witem_id>
<i_performer_flag>
    <item>0</item>
</i_performer_flag>
<r_target_act_seqno>
    <item>-1</item>
</r_target_act_seqno>
<i_is_replica>false</i_is_replica>
<i_vstamp>1</i_vstamp>
<r_object_id>4d0000018000010a</r_object_id>
</properties>
<links>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a"
        rel="self"/>
    <link href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e"
        rel="http://identifiers.emc.com/linkrel/workflow-template"/>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9"
        rel="http://identifiers.emc.com/linkrel/workflow-activity"
        title="Activity-1"/>
</links>
</workflow>

```



#### Example 2-443: JSON response

```
{
    "name": "workflow",
    "type": "dm_workflow",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_workflow",
    "properties": {
        "object_name": "a_test_workflow 2019-02-26",
        "process_id": "4b0000018000236e",
        "r_creator_name": "dmadmin",
        "supervisor_name": "dmadmin",
        "r_runtime_state": 1,
        "r_start_date": "2019-02-26T06:35:58.000+00:00",
        "i_next_seqno": 1,
        "r_alias_set_id": "0000000000000000",
        "parent_id": "0000000000000000",
        "parent_act_seqno": 0,
        "correlation_identifier": "9f000661a430d930db3138306730303f",
        "i_next_act_group_seqno": 0,
        "r_act_seqno": [
            0
        ],
        "r_act_name": [
            "Activity-1"
        ],
    }
}
```

```

    "r_act_def_id": [
        "4c000001800025c9"
    ],
    "r_act_errno": [
        0
    ],
    "r_act_state": [
        1
    ],
    "r_repeat_invoke": [
        true
    ],
    "r_trigger_thresh": [
        0
    ],
    "r_trigger_input": [
        0
    ],
    "r_trigger_revert": [
        0
    ],
    "r_pre_timer": [
        null
    ],
    "r_post_timer": [
        null
    ],
    "r_total_witem": [
        1
    ],
    "r_complete_witem": [
        0
    ],
    "r_last_performer": [
        ""
    ],
    "r_last_witem_id": [
        "0000000000000000"
    ],
    "i_performer_flag": [
        0
    ],
    "r_target_act_seqno": [
        -1
    ],
    "i_is_replica": false,
    "i_vstamp": 1,
    "r_object_id": "4d0000018000010a"
},
"links": [
    {
        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/4d0000018000010a"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-template",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates/4b0000018000236e"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-1",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-activities/4c000001800025c9"
    }
]
}

```



 Example 2-444: HAL + JSON Response

```
{  
    "name": "workflow",  
    "type": "dm_workflow",  
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/  
                  dm_workflow",  
    "properties": {  
        "object_name": "a_test_workflow 2019-02-26",  
        "process_id": "4b0000018000236e",  
        "r_creator_name": "dmadmin",  
        "supervisor_name": "dmadmin",  
        "r_runtime_state": 1,  
        "r_start_date": "2019-02-26T06:35:58.000+00:00",  
        "i_next_seqno": 1,  
        "r_alias_set_id": "0000000000000000",  
        "parent_id": "0000000000000000",  
        "parent_act_seqno": 0,  
        "correlation_identifier": "9f000661a430d930db3138306730303f",  
        "i_next_act_group_seqno": 0,  
        "r_act_seqno": [  
            0  
        ],  
        "r_act_name": [  
            "Activity-1"  
        ],  
        "r_act_def_id": [  
            "4c000001800025c9"  
        ],  
        "r_act_errno": [  
            0  
        ],  
        "r_act_state": [  
            1  
        ],  
        "r_repeat_invoke": [  
            true  
        ],  
        "r_trigger_thresh": [  
            0  
        ],  
        "r_trigger_input": [  
            0  
        ],  
        "r_trigger_revert": [  
            0  
        ],  
        "r_pre_timer": [  
            null  
        ],  
        "r_post_timer": [  
            null  
        ],  
        "r_total_witem": [  
            1  
        ],  
        "r_complete_witem": [  
            0  
        ],  
        "r_last_performer": [  
            ""  
        ],  
        "r_last_witem_id": [  
            "0000000000000000"  
        ],  
        "i_performer_flag": [  
            0  
        ],  
        "r_target_act_seqno": [  
    }  
}
```

```

        - 1
    ],
    "i_is_replica": false,
    "i_vstamp": 1,
    "r_object_id": "4d0000018000010a"
},
"_links": {
    "http://identifiers.emc.com/linkrel/workflow-activity": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9",
        "title": "Activity-1"
    },
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d0000018000010a"
    },
    "http://identifiers.emc.com/linkrel/workflow-template": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e"
    }
}
}

```



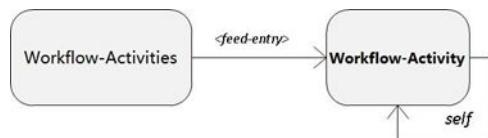
## 2.90 Workflow Activity (ies)

### 2.90.1 Workflow Activity

The *Workflow Activity* resource is used to retrieve a *Workflow Activity* from the *Workflow Activities* resource.

#### 2.90.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Workflow Activities](#)” on page 946 and [About the Diagrams](#) on page 13.

### 2.90.1.2 Feed

No feed support.

### 2.90.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">"Workflow Activity" on page 935</a>

### 2.90.1.4 Operations

#### 2.90.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieve a Workflow Activity.

#### 2.90.1.4.2 Get a Workflow Activity

This operation retrieves a *Workflow Activity*.

##### 2.90.1.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.90.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.90.1.4.2.3 Request media types

N/A

**2.90.1.4.2.4 Request body**

N/A

**2.90.1.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.90.1.4.2.6 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.90.1.4.2.7 Response media types**

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

**2.90.1.4.2.8 Response body**

 **Example 2-445: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow-activity
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_activity"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_activity">
<properties>
    <object_name>Activity-1</object_name>
    <r_object_type>dm_activity</r_object_type>
    <r_creation_date>2018-12-25T09:17:29.000+00:00</r_creation_date>
    <r_modify_date>2018-12-25T09:17:30.000+00:00</r_modify_date>
    <r_modifier>dmadmin</r_modifier>
    <a_is_hidden>false</a_is_hidden>
    <i_is_deleted>false</i_is_deleted>
    <a_archive>false</a_archive>
    <a_link_resolved>false</a_link_resolved>
    <i_reference_cnt>1</i_reference_cnt>
    <i_has_folder>true</i_has_folder>
    <i_folder_id>
        <item>0b00000180002364</item>
    </i_folder_id>
    <r_link_cnt>0</r_link_cnt>
    <r_link_high_cnt>0</r_link_high_cnt>
    <r_assembled_from_id>0000000000000000</r_assembled_from_id>
    <r_frzn_assembly_cnt>0</r_frzn_assembly_cnt>
```

```
<r_has_frzn_assembly>false</r_has_frzn_assembly>
<r_is_virtual_doc>0</r_is_virtual_doc>
<i_contents_id>0000000000000000</i_contents_id>
<r_page_cnt>0</r_page_cnt>
<r_content_size>0</r_content_size>
<a_full_text>true</a_full_text>
<i_cabinet_id>0c00000180002114</i_cabinet_id>
<owner_name>dmadmin</owner_name>
<owner_permit>7</owner_permit>
<group_name>docu</group_name>
<group_permit>5</group_permit>
<world_permit>3</world_permit>
<i_antecedent_id>0000000000000000</i_antecedent_id>
<i_chronicle_id>4c000001800025c9</i_chronicle_id>
<i_latest_flag>true</i_latest_flag>
<r_version_label>
    <item>1.0</item>
    <item>CURRENT</item>
</r_version_label>
<i_branch_cnt>0</i_branch_cnt>
<i_direct_dsc>false</i_direct_dsc>
<r_immutable_flag>false</r_immutable_flag>
<r_frozen_flag>false</r_frozen_flag>
<r_has_events>false</r_has_events>
<acl_domain>dmadmin</acl_domain>
<acl_name>dm_4500000180000914</acl_name>
<a_special_app>palette/activity.gif</a_special_app>
<i_is_reference>false</i_is_reference>
<r_creator_name>dmadmin</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_is_template>false</a_is_template>
<r_full_content_size>0.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>5</i_vstamp>
<is_private>false</is_private>
<r_definition_state>2</r_definition_state>
<trigger_threshold>1</trigger_threshold>
<pre_timer>0</pre_timer>
<post_timer>0</post_timer>
<repeatable_invoke>true</repeatable_invoke>
<exec_type>0</exec_type>
<exec_method_id>0000000000000000</exec_method_id>
<exec_save_results>false</exec_save_results>
<exec_time_out>0</exec_time_out>
<exec_err_handling>0</exec_err_handling>
<performer_type>0</performer_type>
<performer_flag>0</performer_flag>
<transition_type>0</transition_type>
<r_condition_id>0000000000000000</r_condition_id>
<sign_off_required>false</sign_off_required>
<resolve_type>0</resolve_type>
<control_flag>0</control_flag>
<transition_max_output_cnt>0</transition_max_output_cnt>
<transition_eval_cnt>0</transition_eval_cnt>
<transition_flag>0</transition_flag>
<pre_timer_repeat_last>0</pre_timer_repeat_last>
<pre_timer_calendar_flag>0</pre_timer_calendar_flag>
<pre_timer_calendar_id>0000000000000000</pre_timer_calendar_id>
<post_timer_repeat_last>0</post_timer_repeat_last>
<post_timer_calendar_flag>0</post_timer_calendar_flag>
<post_timer_calendar_id>0000000000000000</post_timer_calendar_id>
<r_performer_cond_id>0000000000000000</r_performer_cond_id>
<exec_retry_max>0</exec_retry_max>
<exec_retry_interval>0</exec_retry_interval>
```

```
<exec_subtype>0</exec_subtype>
<activity_group_id>0000000000000000</activity_group_id>
<activity_group_flag>0</activity_group_flag>
<lsm_id>0000000000000000</lsm_id>
<performer_name>
  <item>N/A</item>
</performer_name>
<r_port_name>
  <item>Input:0</item>
  <item>Input:1</item>
  <item>Input:1</item>
  <item>Output:2</item>
  <item>Output:1</item>
  <item>Output:0</item>
</r_port_name>
<r_port_type>
  <item>INPUT</item>
  <item>REVERT</item>
  <item>REVERT</item>
  <item>OUTPUT</item>
  <item>OUTPUT</item>
  <item>OUTPUT</item>
</r_port_type>
<r_package_name>
  <item/>
  <item>Package0</item>
  <item>Package1</item>
  <item>Package0</item>
  <item/>
  <item>Package0</item>
</r_package_name>
<r_package_type>
  <item/>
  <item>dm_sysobject</item>
  <item>dm_sysobject</item>
  <item>dm_sysobject</item>
  <item/>
  <item>dm_sysobject</item>
</r_package_type>
<r_package_oprtn>
  <item/>
  <item/>
  <item/>
  <item/>
  <item/>
</r_package_oprtn>
<r_package_id>
  <item>0000000000000000</item>
  <item>0000000000000000</item>
  <item>0000000000000000</item>
  <item>0000000000000000</item>
  <item>0000000000000000</item>
  <item>0000000000000000</item>
</r_package_id>
<r_package_label>
  <item/>
  <item>CURRENT</item>
  <item>CURRENT</item>
  <item>CURRENT</item>
  <item/>
  <item>CURRENT</item>
</r_package_label>
<r_package_flag>
  <item></item>
  <item>1</item>
  <item>1</item>
  <item>1</item>
  <item>1</item>
  <item>1</item>
```

```

    </r_package_flag>
    <r_package_report_flag>
        <item>0</item>
        <item>0</item>
        <item>0</item>
        <item>0</item>
        <item>0</item>
    </r_package_report_flag>
    <r_object_id>4c000001800025c9</r_object_id>
</properties>
<links>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9" rel="self"/>
</links>
</workflow-activity>

```



#### ➡ Example 2-446: JSON response

```
{
    "name": "workflow-activity",
    "type": "dm_activity",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_activity",
    "properties": {
        "object_name": "Activity-1",
        "r_object_type": "dm_activity",
        "r_creation_date": "2018-12-25T09:17:29.000+00:00",
        "r_modify_date": "2018-12-25T09:17:30.000+00:00",
        "r_modifier": "dmadmin",
        "a_is_hidden": false,
        "i_is_deleted": false,
        "a_archive": false,
        "a_link_resolved": false,
        "i_reference_cnt": 1,
        "i_has_folder": true,
        "i_folder_id": [
            "0b00000180002364"
        ],
        "r_link_cnt": 0,
        "r_link_high_cnt": 0,
        "r_assembled_from_id": "0000000000000000",
        "r_frzn_assembly_cnt": 0,
        "r_has_frzn_assembly": false,
        "r_is_virtual_doc": 0,
        "i_contents_id": "0000000000000000",
        "r_page_cnt": 0,
        "r_content_size": 0,
        "a_full_text": true,
        "i_cabinet_id": "0c00000180002114",
        "owner_name": "dmadmin",
        "owner_permit": 7,
        "group_name": "docu",
        "group_permit": 5,
        "world_permit": 3,
        "i_antecedent_id": "0000000000000000",
        "i_chronicle_id": "4c000001800025c9",
        "i_latest_flag": true,
        "r_version_label": [
            "1.0",
            "CURRENT"
        ],
        "i_branch_cnt": 0,
        "i_direct_dsc": false,
        "r_immutable_flag": false,
    }
}
```

```

    "r_frozen_flag": false,
    "r_has_events": false,
    "acl_domain": "dmadmin",
    "acl_name": "dm_4500000180000914",
    "a_special_app": "palette/activity.gif",
    "i_is_reference": false,
    "r_creator_name": "dmadmin",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 0.0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 5,
    "is_private": false,
    "r_definition_state": 2,
    "trigger_threshold": 1,
    "pre_timer": 0,
    "post_timer": 0,
    "repeatable_invoke": true,
    "exec_type": 0,
    "exec_method_id": "0000000000000000",
    "exec_save_results": false,
    "exec_time_out": 0,
    "exec_err_handling": 0,
    "performer_type": 0,
    "performer_flag": 0,
    "transition_type": 0,
    "r_condition_id": "0000000000000000",
    "sign_off_required": false,
    "resolve_type": 0,
    "control_flag": 0,
    "transition_max_output_cnt": 0,
    "transition_eval_cnt": 0,
    "transition_flag": 0,
    "pre_timer_repeat_last": 0,
    "pre_timer_calendar_flag": 0,
    "pre_timer_calendar_id": "0000000000000000",
    "post_timer_repeat_last": 0,
    "post_timer_calendar_flag": 0,
    "post_timer_calendar_id": "0000000000000000",
    "r_performer_cond_id": "0000000000000000",
    "exec_retry_max": 0,
    "exec_retry_interval": 0,
    "exec_subtype": 0,
    "activity_group_id": "0000000000000000",
    "activity_group_flag": 0,
    "lsm_id": "0000000000000000",
    "performer_name": [
        "N/A"
    ],
    "r_port_name": [
        "Input:0",
        "Input:1",
        "Input:1",
        "Output:2",
        "Output:1",
        "Output:0"
    ],
    "r_port_type": [
        "INPUT",
        "REVERT",
        "REVERT",
        "OUTPUT",
        "OUTPUT",
        "OUTPUT"
    ]
}

```

```
  ],
  "r_package_name": [
    "",
    "Package0",
    "Package1",
    "Package0",
    "",
    "Package0"
  ],
  "r_package_type": [
    "",
    "dm_sysobject",
    "dm_sysobject",
    "dm_sysobject",
    "",
    "dm_sysobject"
  ],
  "r_package_oprtn": [
    "",
    "",
    "",
    "",
    "",
    ""
  ],
  "r_package_id": [
    "0000000000000000",
    "0000000000000000",
    "0000000000000000",
    "0000000000000000",
    "0000000000000000",
    "0000000000000000"
  ],
  "r_package_label": [
    "",
    "CURRENT",
    "CURRENT",
    "CURRENT",
    "",
    "CURRENT"
  ],
  "r_package_flag": [
    1,
    1,
    1,
    1,
    1,
    1
  ],
  "r_package_report_flag": [
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "r_object_id": "4c000001800025c9"
},
"_links": {
  "self": {
    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
    workflow-activities/4c000001800025c9"
  }
}
```



▶ **Example 2-447: HAL: JSON Response**

```
{
  "name": "workflow-template",
  "type": "dm_process",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
    dm_process",
  "properties": {
    "object_name": "a_test_workflow",
    "r_object_type": "dm_process",
    "r_creation_date": "2018-12-25T09:17:22.000+00:00",
    "r_modify_date": "2018-12-25T09:17:32.000+00:00",
    "r_modifier": "dmadmin",
    "a_is_hidden": false,
    "i_is_deleted": false,
    "a_archive": false,
    "a_link_resolved": false,
    "i_reference_cnt": 1,
    "i_has_folder": true,
    "i_folder_id": [
      "0b00000180002364"
    ],
    "r_link_cnt": 0,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "0000000000000000",
    "r_frnz_assembly_cnt": 0,
    "r_has_frnz_assembly": false,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0600000180001500",
    "a_content_type": "dm_internal",
    "r_page_cnt": 1,
    "r_content_size": 2525,
    "a_full_text": true,
    "a_storage_type": "filestore_01",
    "i_cabinet_id": "0c00000180002114",
    "owner_name": "dmadmin",
    "owner_permit": 7,
    "group_name": "docu",
    "group_permit": 5,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "4b0000018000236e",
    "i_latest_flag": true,
    "r_version_label": [
      "1.0",
      "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": false,
    "r_frozen_flag": false,
    "r_has_events": false,
    "acl_domain": "dmadmin",
    "acl_name": "dm_4500000180000914",
    "i_is_reference": false,
    "r_creator_name": "dmadmin",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 2525.0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 4,
    "is_private": false,
    "r_definition_state": 2,
  }
}
```

```
"perf_alias_set_id": "0000000000000000",
"package_control": 0,
"sd_element_default_acl": "0000000000000000",
"calendar_id": "0000000000000000",
"execution_flag": 0,
"r_act_name": [
    "Auto-Activity-2",
    "Auto-Activity-1",
    "Activity-4",
    "Activity-3",
    "Activity-2",
    "Activity-1"
],
"r_act_priority": [
    0,
    0,
    0,
    0,
    0,
    0
],
"r_act_type": [
    0,
    0,
    0,
    2,
    0,
    1
],
"r_act_def_id": [
    "4c000001800025c4",
    "4c000001800025c5",
    "4c000001800025c6",
    "4c000001800025c7",
    "4c000001800025c8",
    "4c000001800025c9"
],
"r_link_name": [
    "L20",
    "L16",
    "L15",
    "L14",
    "L11",
    "L10",
    "L27",
    "L26",
    "L19",
    "L30"
],
"r_link_src_act": [
    "Auto-Activity-2",
    "Activity-4",
    "Activity-2",
    "Activity-2",
    "Activity-2",
    "Activity-1",
    "Activity-1",
    "Auto-Activity-1",
    "Activity-4"
],
"r_link_src_port": [
    "Output:0",
    "Output:0",
    "Output:2",
    "Output:1",
    "Output:0",
    "Output:0",
    "Output:2",
    "Output:1",
    "Output:0"
]
```

```

        "Output:0",
        "Output:1"
    ],
    "r_link_dest_act": [
        "Activity-3",
        "Activity-3",
        "Activity-4",
        "Activity-1",
        "Activity-3",
        "Activity-2",
        "Activity-4",
        "Auto-Activity-1",
        "Auto-Activity-2",
        "Activity-4"
    ],
    "r_link_dest_port": [
        "Input:2",
        "Input:1",
        "Input:0",
        "Input:1",
        "Input:0",
        "Input:0",
        "Input:1",
        "Input:0",
        "Input:0",
        "Input:2"
    ],
    "r_object_id": "4b0000018000236e"
},
"_links": {
    "http://identifiers.emc.com/linkrel/workflow-activity": [
        {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c4",
            "title": "Auto-Activity-2"
        },
        {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c5",
            "title": "Auto-Activity-1"
        },
        {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c6",
            "title": "Activity-4"
        },
        {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c7",
            "title": "Activity-3"
        },
        {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c8",
            "title": "Activity-2"
        },
        {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c9",
            "title": "Activity-1"
        }
    ],
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-templates/4b0000018000236e"
    },
    "http://identifiers.emc.com/linkrel/workflow-activities": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-templates/4b0000018000236e/activities"
    }
}
}
}

```



## 2.90.2 Workflow Activities

The *Workflow Activities* resource retrieves a list of *Workflow Activities*.

### 2.90.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Workflow Template” on page 959 and About the Diagrams on page 13.

### 2.90.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Workflow activities
- Feed Updated: Server's current time
- Supports POST: No
- Entry: [Workflow Activity](#) resource
- Entry ID: The entry URI
- Entry Title: The name of the workflow activity
- Entry Updated: The last modify date

### 2.90.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	“Workflow Activities” on page 946

## 2.90.2.4 Operations

### 2.90.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of Workflow Activity templates as a feed.

### 2.90.2.4.2 Get Workflow template activities

This operation retrieves a collection of *Workflow Activities* from a *Workflow Template* instance.

#### 2.90.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, sort, view, filter, inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			



**Note:** The *page*, *items-per-page*, and *include-total* parameters are not supported.

#### 2.90.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.90.2.4.2.3 Request media types

N/A

#### 2.90.2.4.2.4 Request body

N/A

#### 2.90.2.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.90.2.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.90.2.4.2.7 Response media types

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.90.2.4.2.8 Response body

##### Example 2-448: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates/
        4b0000018000236e/activities</id>
    <title>Workflow activities</title>
    <author>
        <name>OpenText Documentum</name>
    </author>
    <updated>2019-02-25T08:42:04.881+00:00</updated>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e/activities" rel="self"/>
    <entry>
        <id>http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9</id>
        <title>Activity-1</title>
        <summary>dm_activity 4c000001800025c9</summary>
        <updated>2018-12-25T09:17:30.000+00:00</updated>
        <published>2018-12-25T09:17:29.000+00:00</published>
        <link href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9"
            rel="edit"/>
        <content src="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9"
            type="application/vnd.emc.documentum+xml"/>
    </entry>
</feed>
```



▶ **Example 2-449: JSON response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e/activities",
  "title": "Workflow activities",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-25T08:45:29.341+00:00",
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-templates/4b0000018000236e/activities"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9",
      "title": "Activity-1",
      "summary": "dm_activity 4c000001800025c9",
      "updated": "2018-12-25T09:17:30.000+00:00",
      "published": "2018-12-25T09:17:29.000+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                  workflow-activities/4c000001800025c9"
        }
      ],
      "content": {
        "type": "application/vnd.emc.documentum+json",
        "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-activities/4c000001800025c9"
      }
    }
  ]
}
```



▶ **Example 2-450: HAL: JSON Response**

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e/activities",
  "title": "Workflow activities",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-25T08:45:30.654+00:00",
  "_embedded": {
    "collection": [
      {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-activities/4c000001800025c9",
        "title": "Activity-1",
        "summary": "dm_activity 4c000001800025c9",
        "updated": "2018-12-25T09:17:30.000+00:00",
        "published": "2018-12-25T09:17:29.000+00:00",
        "links": {
          "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c9"
          }
        }
      }
    ]
  }
}
```

```
    ],
    "_links": {
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-templates/4b0000018000236e/activities"
        }
    }
}
```



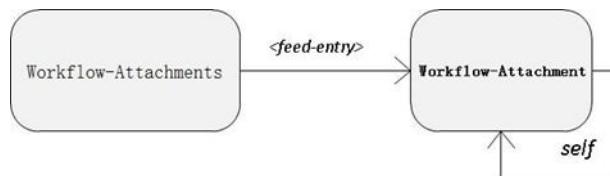
## 2.91 Workflow Attachment(s)

### 2.91.1 Workflow Attachment

The *Workflow Attachment* resource is used to remove a specified *Workflow Attachment*.

#### 2.91.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Workflow Attachments” on page 952 and About the Diagrams on page 13.

#### 2.91.1.2 Feed

No feed support.

#### 2.91.1.3 Link relations

N/A

#### 2.91.1.4 Operations

#### 2.91.1.4.1 Supported HTTP methods

Method	Description
DELETE	Removes a specified <i>Workflow Attachment</i> .

#### 2.91.1.4.2 Remove a Workflow Attachment

This operation removes a specified *Workflow Attachment*.

##### 2.91.1.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
DELETE		For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

##### 2.91.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

##### 2.91.1.4.2.3 Request media types

N/A

##### 2.91.1.4.2.4 Request body

N/A

##### 2.91.1.4.2.5 Response status

- 204 - Removed successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.91.1.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.91.1.4.2.7 Response media types

N/A

#### 2.91.1.4.2.8 Response body

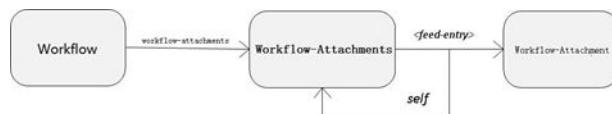
N/A

## 2.91.2 Workflow Attachments

This resource is used to retrieve a collection of *Workflow Attachments*.

### 2.91.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see [“Workflow” on page 906](#) and [About the Diagrams on page 13](#).

### 2.91.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: *Workflow Attachments*
- Feed Updated: Server's current time
- Supports POST: Yes
- Entry: *Workflow Attachment* resource
- Entry ID: The entry URI
- Entry Title: The name of the attachment
- Entry Updated: The last modify date

### 2.91.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	"Workflow Attachments" on page 952

### 2.91.2.4 Operations

#### 2.91.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves the workflow attachments.
POST	Adds an attachment to a <i>Workflow</i> .

#### 2.91.2.4.2 Get Workflow attachments

This operation retrieves workflow attachments.

##### 2.91.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters</a> on page 1047.			

 **Note:** Please note, *page*, *items-per-page*, *include-total*, *sort*, *view*, and *filter* parameters are not supported.

##### 2.91.2.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

##### 2.91.2.4.2.3 Request media types

N/A

**2.91.2.4.2.4 Request body**

N/A

**2.91.2.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

**2.91.2.4.2.6 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

**2.91.2.4.2.7 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

**2.91.2.4.2.8 Response body****Example 2-451: XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id>http://localhost:8080/dctm-rest/repositories/REPO/workflows/
    4d0000018000010a/attachments</id>
  <title>Workflow attachments</title>
  <author>
    <name>OpenText Documentum</name>
  </author>
  <updated>2019-02-26T09:19:49.144+00:00</updated>
  <link
    href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
      4d0000018000010a/attachments" rel="self"/>
  <entry>
    <id>http://localhost:8080/dctm-rest/repositories/REPO/workflows/
      4d0000018000010a/attachments/0000000180000554</id>
    <title>TargetSetup.Result</title>
    <author>
      <name>dmadmin</name>
    </author>
    <summary>dm_sysobject</summary>
    <updated>2019-02-26T09:19:49.144+00:00</updated>
    <published>2019-02-26T08:32:51.000+00:00</published>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments/0000000180000554"
      rel="http://identifiers.emc.com/linkrel/delete"/>
```

```

<content
    src="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments/0000000180000554"
    type="application/vnd.emc.documentum+xml"/>
</entry>
</feed>

```



#### Example 2-452: JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments",
    "title": "Workflow attachments",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-26T09:19:54.624+00:00",
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d0000018000010a/attachments"
        }
    ],
    "entries": [
        {
            "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d0000018000010a/attachments/0000000180000554",
            "title": "TargetSetup.Result",
            "author": [
                {
                    "name": "dmadmin"
                }
            ],
            "summary": "dm_sysobject",
            "updated": "2019-02-26T09:19:54.624+00:00",
            "published": "2019-02-26T08:32:51.000+00:00",
            "links": [
                {
                    "rel": "http://identifiers.emc.com/linkrel/delete",
                    "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                        workflows/4d0000018000010a/attachments/0000000180000554"
                }
            ],
            "content": {
                "type": "application/vnd.emc.documentum+json",
                "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflows/4d0000018000010a/attachments/0000000180000554"
            }
        }
    ]
}
```



#### Example 2-453: HAL+JSON response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments",
    "title": "Workflow attachments",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-26T09:19:57.162+00:00",
    "_embedded": {

```

```

"collection": [
    {
        "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d0000018000010a/attachments/0000000180000554",
        "title": "TargetSetup.Result",
        "author": [
            {
                "name": "dmadmin"
            }
        ],
        "summary": "dm_sysobject",
        "updated": "2019-02-26T09:19:57.162+00:00",
        "published": "2019-02-26T08:32:51.000+00:00",
        "_links": {
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflows/4d0000018000010a/attachments/
                    0000000180000554"
            },
            "http://identifiers.emc.com/linkrel/delete": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflows/4d0000018000010a/attachments/
                    0000000180000554"
            }
        }
    }
],
"_links": {
    "self": {
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
            4d0000018000010a/attachments"
    }
}
}

```



#### 2.91.2.4.3 Add attachment to a Workflow

This operation adds an attachment to a *Workflow*.

##### 2.91.2.4.3.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
POST		Appendix B, REST Common Definition - URI Request Query Parameters on page 1047			

##### 2.91.2.4.3.2 Request headers

- Authorization
- Accept
- Content-Type

For more information, see Appendix D, REST Common Definition - HTTP Headers on page 1055.

#### 2.91.2.4.3.3 Request media types

- application/vnd.emc.documentum+xm
- application/vnd.emc.documentum+json
- application/hal+json

#### 2.91.2.4.3.4 Request body

##### ➤ Example 2-454: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<attachment>
  <component-id>0900000180000220</component-id>
  <component-type>dm_sysobject</component-type>
</attachment>
```



##### ➤ Example 2-455: JSON request

```
{
  "component-id": "0900000180000220",
  "component-type": "dm_sysobject"
}
```



##### ➤ Example 2-456: JSON + HAL request

```
{
  "component-id": "0900000180000220",
  "component-type": "dm_sysobject"
}
```



#### 2.91.2.4.3.5 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

#### 2.91.2.4.3.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.91.2.4.3.7 Response media types

- application/vnd.emc.documentum+xm
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.91.2.4.3.8 Response body

##### Example 2-457: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<attachment xmlns:dm="http://identifiers.emc.com/vocab/documentum">
    <object-id>0000000180000554</object-id>
    <component-id>0900000180000220</component-id>
    <component-name>TargetSetup.Result</component-name>
    <component-type>dm_sysobject</component-type>
    <creator-name>dmadmin</creator-name>
    <creation-date>2019-02-26T08:32:51.000+00:00</creation-date>
    <workflow-id>4d0000018000010a</workflow-id>
    <links>
        <link
            href="http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d0000018000010a/attachments/0000000180000554"
            rel="http://identifiers.emc.com/linkrel/delete"/>
    </links>
</attachment>
```



##### Example 2-458: JSON response

```
{
    "object-id": "0000000180000554",
    "component-id": "0900000180000220",
    "component-name": "TargetSetup.Result",
    "component-type": "dm_sysobject",
    "creator-name": "dmadmin",
    "creation-date": "2019-02-26T08:32:51.000+00:00",
    "workflow-id": "4d0000018000010a",
    "links": [
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
                4d0000018000010a/attachments/0000000180000554"
        }
    ]
}
```



▶ **Example 2-459: HAL+JSON response**

```
{
  "object-id": "0000000180000554",
  "component-id": "0900000180000220",
  "component-name": "TargetSetup.Result",
  "component-type": "dm_sysobject",
  "creator-name": "dmadmin",
  "creation-date": "2019-02-26T08:32:51.000+00:00",
  "workflow-id": "4d0000018000010a",
  "_links": {
    "http://identifiers.emc.com/linkrel/delete": {
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/workflows/
        4d0000018000010a/attachments/0000000180000554"
    }
  }
}
```



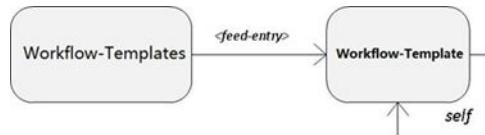
## 2.92 Workflow Template(s)

### 2.92.1 Workflow Template

The Workflow Template resource is used to get a specified Workflow Template.

#### 2.92.1.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “Workflow Templates” on page 970 and About the Diagrams on page 13.

#### 2.92.1.2 Feed

No feed support.

#### 2.92.1.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	“Workflow Template” on page 959
workflow-activities [1]	Link to a specific <i>Workflow Activities</i> collection	“Workflow Activities” on page 946
workflow-activity [1]	Link to a specific <i>Workflow Activity</i>	“Workflow Activity” on page 935

Link relation	Description	Resource reference
[1] This link relation is defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .		

## 2.92.1.4 Operations

### 2.92.1.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a <i>Workflow Template</i>

### 2.92.1.4.2 Get Workflow Template

This operation retrieves a specified *Workflow Template*.

#### 2.92.1.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	links, view	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### 2.92.1.4.2.2 Request headers

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.92.1.4.2.3 Request media types

N/A

#### 2.92.1.4.2.4 Request body

N/A

#### 2.92.1.4.2.5 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.92.1.4.2.6 Response headers

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.92.1.4.2.7 Response media types

- application/vnd.emc.documentum+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.92.1.4.2.8 Response body

##### ➡ Example 2-460: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow-template
    definition="http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_process"
    xmlns="http://identifiers.emc.com/vocab/documentum"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="dm_process">
    <properties>
        <object_name>a_test_workflow</object_name>
        <r_object_type>dm_process</r_object_type>
        <r_creation_date>2018-12-25T09:17:22.000+00:00</r_creation_date>
        <r_modify_date>2018-12-25T09:17:32.000+00:00</r_modify_date>
        <r_modifier>dmadmin</r_modifier>
        <a_is_hidden>false</a_is_hidden>
        <i_is_deleted>false</i_is_deleted>
        <a_archive>false</a_archive>
        <a_link_resolved>false</a_link_resolved>
        <i_reference_cnt>1</i_reference_cnt>
        <i_has_folder>true</i_has_folder>
        <i_folder_id>
            <item>0b00000180002364</item>
        </i_folder_id>
        <r_link_cnt>0</r_link_cnt>
        <r_link_high_cnt>0</r_link_high_cnt>
        <r_assembled_from_id>0000000000000000</r_assembled_from_id>
        <r_frzn_assembly_cnt>0</r_frzn_assembly_cnt>
        <r_has_frzn_assembly>false</r_has_frzn_assembly>
        <r_is_virtual_doc>0</r_is_virtual_doc>
        <i_contents_id>0600000180001500</i_contents_id>
        <a_content_type>dm_internal</a_content_type>
        <r_page_cnt>1</r_page_cnt>
        <r_content_size>2525</r_content_size>
```

```
<a_full_text>true</a_full_text>
<a_storage_type>filestore_01</a_storage_type>
<i_cabinet_id>0c00000180002114</i_cabinet_id>
<owner_name>dmadmin</owner_name>
<owner_permit>7</owner_permit>
<group_name>docu</group_name>
<group_permit>5</group_permit>
<world_permit>3</world_permit>
<i_antecedent_id>0000000000000000</i_antecedent_id>
<i_chronicle_id>4b0000018000236e</i_chronicle_id>
<i_latest_flag>true</i_latest_flag>
<r_version_label>
    <item>1.0</item>
    <item>CURRENT</item>
</r_version_label>
<i_branch_cnt>0</i_branch_cnt>
<i_direct_dsc>false</i_direct_dsc>
<r_immutable_flag>false</r_immutable_flag>
<r_frozen_flag>false</r_frozen_flag>
<r_has_events>false</r_has_events>
<acl_domain>dmadmin</acl_domain>
<acl_name>dm_4500000180000914</acl_name>
<i_is_reference>false</i_is_reference>
<r_creator_name>dmadmin</r_creator_name>
<r_is_public>true</r_is_public>
<r_policy_id>0000000000000000</r_policy_id>
<r_resume_state>0</r_resume_state>
<r_current_state>0</r_current_state>
<r_alias_set_id>0000000000000000</r_alias_set_id>
<a_is_template>false</a_is_template>
<r_full_content_size>2525.0</r_full_content_size>
<a_is_signed>false</a_is_signed>
<i_partition>0</i_partition>
<i_is_replica>false</i_is_replica>
<i_vstamp>4</i_vstamp>
<is_private>false</is_private>
<r_definition_state>2</r_definition_state>
<perf_alias_set_id>0000000000000000</perf_alias_set_id>
<package_control>0</package_control>
<sd_element_default_acl>0000000000000000</sd_element_default_acl>
<calendar_id>0000000000000000</calendar_id>
<execution_flag>0</execution_flag>
<r_act_name>
    <item>Auto-Activity-2</item>
    <item>Auto-Activity-1</item>
    <item>Activity-4</item>
    <item>Activity-3</item>
    <item>Activity-2</item>
    <item>Activity-1</item>
</r_act_name>
<r_act_priority>
    <item>0</item>
    <item>0</item>
    <item>0</item>
    <item>0</item>
    <item>0</item>
    <item>0</item>
</r_act_priority>
<r_act_type>
    <item>0</item>
    <item>0</item>
    <item>0</item>
    <item>2</item>
    <item>0</item>
    <item>1</item>
</r_act_type>
<r_act_def_id>
    <item>4c000001800025c4</item>
    <item>4c000001800025c5</item>
    <item>4c000001800025c6</item>
```

```

        <item>4c000001800025c7</item>
        <item>4c000001800025c8</item>
        <item>4c000001800025c9</item>
    </r_act_def_id>
    <r_link_name>
        <item>L20</item>
        <item>L16</item>
        <item>L15</item>
        <item>L14</item>
        <item>L11</item>
        <item>L10</item>
        <item>L27</item>
        <item>L26</item>
        <item>L19</item>
        <item>L30</item>
    </r_link_name>
    <r_link_src_act>
        <item>Auto-Activity-2</item>
        <item>Activity-4</item>
        <item>Activity-2</item>
        <item>Activity-2</item>
        <item>Activity-2</item>
        <item>Activity-1</item>
        <item>Activity-1</item>
        <item>Activity-1</item>
        <item>Auto-Activity-1</item>
        <item>Activity-4</item>
    </r_link_src_act>
    <r_link_src_port>
        <item>Output:0</item>
        <item>Output:0</item>
        <item>Output:2</item>
        <item>Output:1</item>
        <item>Output:0</item>
        <item>Output:0</item>
        <item>Output:2</item>
        <item>Output:1</item>
        <item>Output:0</item>
        <item>Output:1</item>
    </r_link_src_port>
    <r_link_dest_act>
        <item>Activity-3</item>
        <item>Activity-3</item>
        <item>Activity-4</item>
        <item>Activity-1</item>
        <item>Activity-3</item>
        <item>Activity-2</item>
        <item>Activity-4</item>
        <item>Auto-Activity-1</item>
        <item>Auto-Activity-2</item>
        <item>Activity-4</item>
    </r_link_dest_act>
    <r_link_dest_port>
        <item>Input:2</item>
        <item>Input:1</item>
        <item>Input:0</item>
        <item>Input:1</item>
        <item>Input:0</item>
        <item>Input:1</item>
        <item>Input:0</item>
        <item>Input:0</item>
        <item>Input:2</item>
    </r_link_dest_port>
    <r_object_id>4b0000018000236e</r_object_id>
</properties>
<links>
    <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates/4b0000018000236e" rel="self" />

```

```

<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e/activities"
      rel="http://identifiers.emc.com/linkrel/workflow-activities"/>
<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c4"
      rel="http://identifiers.emc.com/linkrel/workflow-activity"
      title="Auto-Activity-2"/>
<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c5"
      rel="http://identifiers.emc.com/linkrel/workflow-activity"
      title="Auto-Activity-1"/>
<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c6"
      rel="http://identifiers.emc.com/linkrel/workflow-activity"
      title="Activity-4"/>
<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c7"
      rel="http://identifiers.emc.com/linkrel/workflow-activity"
      title="Activity-3"/>
<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c8"
      rel="http://identifiers.emc.com/linkrel/workflow-activity"
      title="Activity-2"/>
<link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9"
      rel="http://identifiers.emc.com/linkrel/workflow-activity"
      title="Activity-1"/>
</links>
</workflow-template>
```



### Example 2-461: JSON response

```
{
  "name": "workflow-template",
  "type": "dm_process",
  "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
    dm_process",
  "properties": {
    "object_name": "a_test_workflow",
    "r_object_type": "dm_process",
    "r_creation_date": "2018-12-25T09:17:22.000+00:00",
    "r_modify_date": "2018-12-25T09:17:32.000+00:00",
    "r_modifier": "dmadmin",
    "a_is_hidden": false,
    "i_is_deleted": false,
    "a_archive": false,
    "a_link_resolved": false,
    "i_reference_cnt": 1,
    "i_has_folder": true,
    "i_folder_id": [
      "0b00000180002364"
    ],
    "r_link_cnt": 0,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "0000000000000000",
    "r_frzn_assembly_cnt": 0,
    "r_has_frzn_assembly": false,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0600000180001500",
    "i_content_type": "application/pdf"
  }
}
```

```

    "a_content_type": "dm_internal",
    "r_page_cnt": 1,
    "r_content_size": 2525,
    "a_full_text": true,
    "a_storage_type": "filestore_01",
    "i_cabinet_id": "0c00000180002114",
    "owner_name": "dmadmin",
    "owner_permit": 7,
    "group_name": "docu",
    "group_permit": 5,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "4b0000018000236e",
    "i_latest_flag": true,
    "r_version_label": [
        "1.0",
        "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": false,
    "r_frozen_flag": false,
    "r_has_events": false,
    "acl_domain": "dmadmin",
    "acl_name": "dm_4500000180000914",
    "i_is_reference": false,
    "r_creator_name": "dmadmin",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 2525.0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 4,
    "is_private": false,
    "r_definition_state": 2,
    "perf_alias_set_id": "0000000000000000",
    "package_control": 0,
    "sd_element_default_acl": "0000000000000000",
    "calendar_id": "0000000000000000",
    "execution_flag": 0,
    "r_act_name": [
        "Auto-Activity-2",
        "Auto-Activity-1",
        "Activity-4",
        "Activity-3",
        "Activity-2",
        "Activity-1"
    ],
    "r_act_priority": [
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "r_act_type": [
        0,
        0,
        0,
        2,
        0,
        1
    ],
    "r_act_def_id": [

```

```
"4c000001800025c4",
"4c000001800025c5",
"4c000001800025c6",
"4c000001800025c7",
"4c000001800025c8",
"4c000001800025c9"
],
"r_link_name": [
    "L20",
    "L16",
    "L15",
    "L14",
    "L11",
    "L10",
    "L27",
    "L26",
    "L19",
    "L30"
],
"r_link_src_act": [
    "Auto-Activity-2",
    "Activity-4",
    "Activity-2",
    "Activity-2",
    "Activity-2",
    "Activity-1",
    "Activity-1",
    "Activity-1",
    "Auto-Activity-1",
    "Activity-4"
],
"r_link_src_port": [
    "Output:0",
    "Output:0",
    "Output:2",
    "Output:1",
    "Output:0",
    "Output:0",
    "Output:2",
    "Output:1",
    "Output:0",
    "Output:1"
],
"r_link_dest_act": [
    "Activity-3",
    "Activity-3",
    "Activity-4",
    "Activity-1",
    "Activity-3",
    "Activity-2",
    "Activity-4",
    "Auto-Activity-1",
    "Auto-Activity-2",
    "Activity-4"
],
"r_link_dest_port": [
    "Input:2",
    "Input:1",
    "Input:0",
    "Input:1",
    "Input:0",
    "Input:0",
    "Input:1",
    "Input:0",
    "Input:0",
    "Input:2"
],
"r_object_id": "4b0000018000236e"
},
"links": [{
```

```

        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activities",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e/activities"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Auto-Activity-2",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c4"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Auto-Activity-1",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c5"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-4",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c6"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-3",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c7"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-2",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c8"
    }, {
        "rel": "http://identifiers.emc.com/linkrel/workflow-activity",
        "title": "Activity-1",
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-activities/4c000001800025c9"
    }
}
]
}

```



#### Example 2-462: HAL: JSON response

```
{
    "name": "workflow-template",
    "type": "dm_process",
    "definition": "http://localhost:8080/dctm-rest/repositories/REPO/types/
        dm_process",
    "properties": {
        "object_name": "a_test_workflow",
        "r_object_type": "dm_process",
        "r_creation_date": "2018-12-25T09:17:22.000+00:00",
        "r_modify_date": "2018-12-25T09:17:32.000+00:00",
        "r_modifier": "dmadmin",
        "a_is_hidden": false,
        "i_is_deleted": false,
        "a_archive": false,
        "a_link_resolved": false,
        "i_reference_cnt": 1,
        "i_has_folder": true,
        "i_folder_id": [
            "0b00000180002364"
        ],
        "r_link_cnt": 0,
        "r_link_high_cnt": 0,
        "r_assembled_from_id": "0000000000000000",
        "r_assembled_to_id": null
    }
}
```

```
"r_frzn_assembly_cnt": 0,
"r_has_frzn_assembly": false,
"r_is_virtual_doc": 0,
"i_contents_id": "0600000180001500",
"a_content_type": "dm_internal",
"r_page_cnt": 1,
"r_content_size": 2525,
"a_full_text": true,
"a_storage_type": "filestore_01",
"i_cabinet_id": "0c00000180002114",
"owner_name": "dmadmin",
"owner_permit": 7,
"group_name": "docu",
"group_permit": 5,
"world_permit": 3,
"i_antecedent_id": "0000000000000000",
"i_chronicle_id": "4b0000018000236e",
"i_latest_flag": true,
"r_version_label": [
    "1.0",
    "CURRENT"
],
"i_branch_cnt": 0,
"i_direct_dsc": false,
"r_immutable_flag": false,
"r_frozen_flag": false,
"r_has_events": false,
"acl_domain": "dmadmin",
"acl_name": "dm_4500000180000914",
"i_is_reference": false,
"r_creator_name": "dmadmin",
"r_is_public": true,
"r_policy_id": "0000000000000000",
"r_resume_state": 0,
"r_current_state": 0,
"r_alias_set_id": "0000000000000000",
"a_is_template": false,
"r_full_content_size": 2525.0,
"a_is_signed": false,
"i_partition": 0,
"i_is_replica": false,
"i_vstamp": 4,
"is_private": false,
"r_definition_state": 2,
"perf_alias_set_id": "0000000000000000",
"package_control": 0,
"sd_element_default_acl": "0000000000000000",
"calendar_id": "0000000000000000",
"execution_flag": 0,
"r_act_name": [
    "Auto-Activity-2",
    "Auto-Activity-1",
    "Activity-4",
    "Activity-3",
    "Activity-2",
    "Activity-1"
],
"r_act_priority": [
    0,
    0,
    0,
    0,
    0,
    0
],
"r_act_type": [
    0,
    0,
    0,
    2,
```

```
        0,
        1
    ],
    "r_act_def_id": [
        "4c000001800025c4",
        "4c000001800025c5",
        "4c000001800025c6",
        "4c000001800025c7",
        "4c000001800025c8",
        "4c000001800025c9"
    ],
    "r_link_name": [
        "L20",
        "L16",
        "L15",
        "L14",
        "L11",
        "L10",
        "L27",
        "L26",
        "L19",
        "L30"
    ],
    "r_link_src_act": [
        "Auto-Activity-2",
        "Activity-4",
        "Activity-2",
        "Activity-2",
        "Activity-2",
        "Activity-1",
        "Activity-1",
        "Activity-1",
        "Auto-Activity-1",
        "Activity-4"
    ],
    "r_link_src_port": [
        "Output:0",
        "Output:0",
        "Output:2",
        "Output:1",
        "Output:0",
        "Output:0",
        "Output:2",
        "Output:1",
        "Output:0",
        "Output:1"
    ],
    "r_link_dest_act": [
        "Activity-3",
        "Activity-3",
        "Activity-4",
        "Activity-1",
        "Activity-3",
        "Activity-2",
        "Activity-4",
        "Auto-Activity-1",
        "Auto-Activity-2",
        "Activity-4"
    ],
    "r_link_dest_port": [
        "Input:2",
        "Input:1",
        "Input:0",
        "Input:1",
        "Input:0",
        "Input:0",
        "Input:1",
        "Input:0",
        "Input:0",
        "Input:2"
    ]
}
```

```

        ],
        "r_object_id": "4b0000018000236e"
    },
    "_links": {
        "http://identifiers.emc.com/linkrel/workflow-activity": [
            {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c4",
                "title": "Auto-Activity-2"
            },
            {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c5",
                "title": "Auto-Activity-1"
            },
            {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c6",
                "title": "Activity-4"
            },
            {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c7",
                "title": "Activity-3"
            },
            {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c8",
                "title": "Activity-2"
            },
            {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                    workflow-activities/4c000001800025c9",
                "title": "Activity-1"
            }
        ],
        "self": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-templates/4b0000018000236e"
        },
        "http://identifiers.emc.com/linkrel/workflow-activities": {
            "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-templates/4b0000018000236e/activities"
        }
    }
}

```



## 2.92.2 Workflow Templates

This resource is used to get a collection of *Workflow Templates*, also called *Processes*.

### 2.92.2.1 Resource relationships

The following diagram illustrates how this resource is related to other resources:



For more information, see “[Repository](#)” on page 564 and [About the Diagrams](#) on page 13.

### 2.92.2.2 Feed

Is this resource a feed? Yes.

- Feed ID: The resource URI
- Feed Title: Workflow Templates
- Feed Updated: Server's current time
- Supports POST: No
- Entry: Workflow Template Resource
- Entry ID: The entry URI
- Entry Title: The name of the Workflow Template
- Entry Updated: The last modify date

### 2.92.2.3 Link relations

Link relation	Description	Resource reference
self	Link to this resource	<a href="#">"Workflow Templates" on page 970</a>

### 2.92.2.4 Operations

#### 2.92.2.4.1 Supported HTTP methods

Method	Description
GET	Retrieves a collection of <i>Workflow Templates</i>

#### 2.92.2.4.2 Get Workflow Templates

This operation retrieves a collection of *Workflow Templates*.

##### 2.92.2.4.2.1 HTTP Request Method and Query Parameters

HTTP method	Query parameters	Description	Data type	Value range	Default value
GET	page, items-per-page, include-total, links, sort, view, filter, inline	For more information, see <a href="#">Appendix B, REST Common Definition - URI Request Query Parameters on page 1047</a> .			

#### **2.92.2.4.2.2 Request headers**

- Authorization
- Accept

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.92.2.4.2.3 Request media types**

N/A

#### **2.92.2.4.2.4 Request body**

N/A

#### **2.92.2.4.2.5 Response status**

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.92.2.4.2.6 Response headers**

- Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.92.2.4.2.7 Response media types**

- application/atom+xml
- application/xml
- application/vnd.emc.documentum+json
- application/json
- application/hal+json

#### 2.92.2.4.2.8 Response body

➡ Example 2-463: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates</id>
    <title>Workflow templates</title>
    <author>
      <name>OpenText Documentum</name>
    </author>
    <updated>2019-02-25T07:15:29.574+00:00</updated>
    <dm:page xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:page>
    <dm:items-per-page xmlns:dm="http://identifiers.emc.com/vocab/documentum">
      100
    </dm:items-per-page>
    <link
      href="http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates"
      rel="self"/>
    <entry>
      <id>http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e</id>
      <title>a_test_workflow</title>
      <summary>dm_process 4b0000018000236e</summary>
      <updated>2018-12-25T09:17:32.000+00:00</updated>
      <published>2018-12-25T09:17:22.000+00:00</published>
      <link
        href="http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-templates/4b0000018000236e" rel="edit"/>
      <content
        src="http://localhost:8080/dctm-rest/repositories/REPO/
              workflow-templates/4b0000018000236e"
        type="application/vnd.emc.documentum+xml"/>
    </entry>
  </feed>
```



➡ Example 2-464: JSON response

```
{
  "id": "http://localhost:8080/dctm-rest/repositories/REPO/
        workflow-templates",
  "title": "Workflow templates",
  "author": [
    {
      "name": "OpenText Documentum"
    }
  ],
  "updated": "2019-02-25T07:07:37.869+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                workflow-templates"
    }
  ],
  "entries": [
    {
      "id": "http://localhost:8080/dctm-rest/repositories/REPO/
            workflow-templates/4b0000018000236e",
      "title": "a_test_workflow",
      "summary": "dm_process 4b0000018000236e",
      "updated": "2018-12-25T09:17:32.000+00:00",
      "published": "2018-12-25T09:17:22.000+00:00",
      "links": [
        {
          "rel": "edit",
        }
      ]
    }
  ]
}
```

```
        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                  workflow-templates/4b0000018000236e"
    }
],
"content": {
    "type": "application/vnd.emc.documentum+json",
    "src": "http://localhost:8080/dctm-rest/repositories/REPO/
                  workflow-templates/4b0000018000236e"
}
}
]
```



#### ➤ Example 2-465: HAL: JSON Response

```
{
    "id": "http://localhost:8080/dctm-rest/repositories/REPO/workflow-templates",
    "title": "Workflow templates",
    "author": [
        {
            "name": "OpenText Documentum"
        }
    ],
    "updated": "2019-02-25T07:23:53.443+00:00",
    "page": 1,
    "items-per-page": 100,
    "_embedded": {
        "collection": [
            {
                "id": "http://localhost:8080/dctm-rest/repositories/REPO/
                          workflow-templates/4b0000018000236e",
                "title": "a_test_workflow",
                "summary": "dm_process 4b0000018000236e",
                "updated": "2018-12-25T09:17:32.000+00:00",
                "published": "2018-12-25T09:17:22.000+00:00",
                "_links": {
                    "self": {
                        "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                          workflow-templates/4b0000018000236e"
                    }
                }
            }
        ],
        "_links": {
            "self": {
                "href": "http://localhost:8080/dctm-rest/repositories/REPO/
                  workflow-templates"
            }
        }
    }
}
```



## 2.93 Workspace Type Definition

### 2.93.1 Create Workspace Type Definition

Creates a Workspace Type Definition.

#### 2.93.1.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition

#### 2.93.1.2 HTTP Request method

POST

#### 2.93.1.3 Request media types

- application/vnd.emc.documentum+json
- application/json

#### 2.93.1.4 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.93.1.5 Request body

The JSON request is stored in the database as content in the dm\_bws\_type\_def type.

The items specified in the properties section in the example request body sample are stored as columns and the remaining are stored as JSON content.

##### ➡ Example 2-466: JSON request

```
{
  "properties": {
    "object_name": "hiring_workspace",
    "locales": ["en", "de"],
    "name_labels": ["Hiring Workspace", "Buchhaltung case"],
    "descriptions": ["This is a description for hiring", "Dies ist eine Beschreibung für die Buchhaltung"],
    "icon_name": "mime_wsjob_opening32"
  },
  "bws-folder-locations": [
    {
      "id": "0b0003e98000750c",
      "descendent": true
    }
  ]
}
```

```

        },
    ],
    "type-attributes": [
        {
            "name": "is_admin",
            "type": "Boolean",
            "locales": ["en", "de"],
            "labels": ["is Admin", "ist Admin"],
            "repeating": false
        },
        {
            "name": "account_number",
            "type": "Integer",
            "locales": ["en", "de"],
            "labels": ["Account Number", "Kontonummer"],
            "repeating": false
        },
        {
            "name": "reference_id",
            "type": "ID",
            "locales": ["en", "de"],
            "labels": ["Reference Id", "Referenz-ID"],
            "repeating": false
        },
        {
            "name": "follow_up_date",
            "type": "Time",
            "locales": ["en", "de"],
            "labels": ["Followup Date", "Folgedatum"],
            "repeating": false
        },
        {
            "name": "version_number",
            "type": "Double",
            "locales": ["en", "de"],
            "labels": ["Version Number", "Versionsnummer"],
            "repeating": false
        }
    ],
    "groups": [
        {
            "group-name": "Hiring Manager",
            "locales": ["en", "de"],
            "group-labels": ["Hiring Manager", "Einstellungsmanager"],
            "descriptions": ["This is a description for Hiring Manager", "Dies ist eine Beschreibung für den Einstellungsleiter."],
            "is-admin-role": true
        },
        {
            "group-name": "Applicant",
            "locales": ["en", "de"],
            "group-labels": ["Applicant", "Applicante"],
            "descriptions": ["This is an applicant role", "Das ist eine Bewerberrolle"]
        }
    ]
}

```



### 2.93.1.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.93.1.7 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.93.1.8 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.93.1.9 Response body

JSON representation.

 **Example 2-467: JSON response**

```
{
    "name": "bws-type-definition",
    "type": "dm_bws_type_def",
    "properties": {
        "r_object_id": "080003e980007db4",
        "object_name": "hiring_workspace",
        "icon_name": "mime_wsjob_opening32",
        "locales": [
            "en",
            "de"
        ],
        "name_labels": [
            "Hiring Workspace",
            "Buchhaltung case"
        ],
        "descriptions": [
            "This is a description for hiring",
            "Dies ist eine Beschreibung für die Buchhaltung"
        ],
        "r_creator_name": "dmadmin",
        "r_creation_date": "2025-08-25T10:04:09.000+00:00",
        "r_modifier": "dmadmin",
        "r_modify_date": "2025-08-25T10:04:09.000+00:00"
    },
    "bws-folder-locations": [
        {
            "location-type": "id-location",
            "id": "0b0003e98000750c",
            "descendent": true
        }
    ]
}
```

```
],
"type-attributes": [
{
  "labels": [
    "is Admin",
    "ist Admin"
  ],
  "locales": [
    "en",
    "de"
  ],
  "name": "is_admin",
  "repeating": false,
  "type": "boolean",
  "length": 0
},
{
  "labels": [
    "Account Number",
    "Kontonummer"
  ],
  "locales": [
    "en",
    "de"
  ],
  "name": "account_number",
  "repeating": false,
  "type": "integer",
  "length": 0
},
{
  "labels": [
    "Reference Id",
    "Referenz-ID"
  ],
  "locales": [
    "en",
    "de"
  ],
  "name": "reference_id",
  "repeating": false,
  "type": "id",
  "length": 0
},
{
  "labels": [
    "Followup Date",
    "Folgedatum"
  ],
  "locales": [
    "en",
    "de"
  ],
  "name": "follow_up_date",
  "repeating": false,
  "type": "time",
  "length": 0
},
{
  "labels": [
    "Version Number",
    "Versionsnummer"
  ]]
```

```

        ],
        "locales": [
            "en",
            "de"
        ],
        "name": "version_number",
        "repeating": false,
        "type": "double",
        "length": 0
    }
],
"groups": [
{
    "item-id": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
    "group-name": "Hiring Manager",
    "locales": [
        "en",
        "de"
    ],
    "descriptions": [
        "This is a description for Hiring Manager",
        "Dies ist eine Beschreibung für den Einstellungsleiter."
    ],
    "group-labels": [
        "Hiring Manager",
        "Einstellungsmanager"
    ],
    "is-admin-role": true
},
{
    "item-id": "808cad06-60ee-47bb-96db-dfd166f4cb83",
    "group-name": "Applicant",
    "locales": [
        "en",
        "de"
    ],
    "descriptions": [
        "This is an applicant role",
        "Das ist eine Bewerberrolle"
    ],
    "group-labels": [
        "Applicant",
        "Applicante"
    ],
    "is-admin-role": false
},
{
    "instance-exists": false,
    "subtype-exists": false,
    "links": [
        {
            "rel": "self",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
        },
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
        }
    ]
}
]

```

```
        "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
    },
{
    "rel": "http://identifiers.emc.com/linkrel/bws-folder-templates",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates"
}
]
```



## 2.93.2 Get Workspace Type Definition

Retrieves a specific Workspace Type Definition.

### 2.93.2.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{r\_object\_id}

### 2.93.2.2 HTTP Request method

GET

### 2.93.2.3 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.93.2.4 Request headers

- Accept
- Authorization

For more information, see Appendix D, REST Common Definition - HTTP Headers on page 1055.

### 2.93.2.5 Request body

N/A

### 2.93.2.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.93.2.7 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.93.2.8 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.93.2.9 Response body

JSON representation.

The response is retrieved from the `dm_bws_type_def` type. The response is extracted from the content. All the required Workspace information is available in the response.

#### ➡ Example 2-468: JSON response

```
{
    "name": "bws-type-definition",
    "type": "dm_bws_type_def",
    "properties": {
        "r_object_id": "080003e980007db4",
        "object_name": "hiring_workspace",
        "icon_name": "mime_wsjob_opening32",
        "locales": [
            "en"
        ],
        "name_labels": [
            "Hiring Workspace",
            "Buchhaltung case"
        ],
        "descriptions": [
            "This is a description for hiring",
            "Dies ist eine Beschreibung für die Buchhaltung"
        ],
        "r_creator_name": "dmadmin",
        "r_creation_date": "2025-08-25T10:04:09.000+00:00",
        "r_update_time": "2025-08-25T10:04:09.000+00:00"
    }
}
```

```
        "r_modifier": "dmadmin",
        "r_modify_date": "2025-08-25T10:18:42.000+00:00"
    },
    "bws-folder-locations": [
        {
            "location-type": "id-location",
            "id": "0b0003e98000750c",
            "descendent": true
        }
    ],
    "type-attributes": [
        {
            "labels": [
                "is Admin",
                "ist Admin"
            ],
            "locales": [
                "en",
                "de"
            ],
            "name": "is_admin",
            "repeating": false,
            "type": "boolean",
            "length": 0
        },
        {
            "labels": [
                "Account Number",
                "Kontonummer"
            ],
            "locales": [
                "en",
                "de"
            ],
            "name": "account_number",
            "repeating": false,
            "type": "integer",
            "length": 0
        },
        {
            "labels": [
                "Reference Id",
                "Referenz-ID"
            ],
            "locales": [
                "en",
                "de"
            ],
            "name": "reference_id",
            "repeating": false,
            "type": "id",
            "length": 0
        },
        {
            "labels": [
                "Followup Date",
                "Folgedatum"
            ],
            "locales": [
                "en",
                "de"
            ]
        },
    ]
}
```

```

        "name": "follow_up_date",
        "repeating": false,
        "type": "time",
        "length": 0
    },
    {
        "labels": [
            "Version Number",
            "Versionsnummer"
        ],
        "locales": [
            "en",
            "de"
        ],
        "name": "version_number",
        "repeating": false,
        "type": "double",
        "length": 0
    }
],
"groups": [
    {
        "item-id": "808cad06-60ee-47bb-96db-dfd166f4cb83",
        "group-name": "Applicant",
        "locales": [
            "en"
        ],
        "descriptions": [
            "This is an applicant role",
            "Das ist eine Bewerberrolle"
        ],
        "group-labels": [
            "Applicant",
            "Applicante"
        ],
        "is-admin-role": false
    },
    {
        "item-id": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
        "group-name": "Hiring Manager",
        "locales": [
            "en"
        ],
        "descriptions": [
            "This is a description for Hiring Manager",
            "Dies ist eine Beschreibung für den Einstellungsleiter."
        ],
        "group-labels": [
            "Hiring Manager",
            "Einstellungsmanager"
        ],
        "is-admin-role": true
    }
],
"permission-templates": [
    {
        "item-id": "f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec",
        "name": "Hiring PT",
        "locales": [
            "en",
            "de"
        ],
        "label": "Hiring PT"
    }
]
}

```

```
        "name-labels": [
            "Hiring PT",
            "Einstellung von Teilzeitkräften"
        ],
        "descriptions": [
            "This is a description for Hiring permission template.",
            "Dies ist eine Beschreibung für die Vorlage zur
Einstellgenehmigung."
        ],
        "permitted": [
            {
                "accessor-type": "USER",
                "accessor-name": "test_user_1",
                "accessor": "110003e980001548",
                "basic-permission": "Write"
            },
            {
                "accessor-type": "GROUP",
                "accessor-name": "admingroup",
                "accessor": "120003e980000d01",
                "basic-permission": "Delete"
            },
            {
                "accessor-type": "BWS",
                "accessor": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
                "basic-permission": "Delete"
            },
            {
                "accessor-type": "BWS",
                "accessor": "808cad06-60ee-47bb-96db-dfd166f4cb83",
                "basic-permission": "Read"
            },
            {
                "accessor-type": "ALIAS",
                "accessor": "dm_owner",
                "basic-permission": "Delete"
            },
            {
                "accessor-type": "ALIAS",
                "accessor": "dm_world",
                "basic-permission": "Read"
            }
        ]
    },
    "instance-exists": false,
    "subtype-exists": false,
    "links": [
        {
            "rel": "self",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
        },
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
        }
    ]
}
```

```

        },
        {
          "rel": "http://identifiers.emc.com/linkrel/bws-folder-templates",
          "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates"
        }
      ]
    }
  
```



### 2.93.3 Get All Workspace Type Definitions

Retrieves all Workspace Type Definitions.

#### 2.93.3.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition?inline=true&links=true

#### 2.93.3.2 HTTP Request method

GET

#### 2.93.3.3 Request media types

- application/vnd.emc.documentum+json
- application/json

#### 2.93.3.4 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.93.3.5 Request body

N/A

### 2.93.3.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.93.3.7 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.93.3.8 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.93.3.9 Response body

JSON representation.

The response is retrieved from the `dm_bws_type_def` type. The response is optimized for the list view and includes only the necessary data required for display. All the Workspace Type Definition information is stored as JSON content and are excluded in the response.

#### Example 2-469: JSON response

```
{  
    "id": "/repositories/d2repo/bws-types-definition",  
    "title": "List of Business Workspaces",  
    "author": [  
        {  
            "name": "Open Text Documentum"  
        }  
    ],  
    "updated": "2025-08-25T10:31:40.435+00:00",  
    "page": 1,  
    "items-per-page": 100,  
    "links": [  
        {  
            "rel": "self",  
            "href": "/repositories/d2repo/bws-types-definition?  
            inline=true&links=true&filter=object_name=%27hirings_workspace%27"  
        }  
    ],  
    "entries": [  
        {  
            "id": "/repositories/d2repo/bws-types-definition/080003e980007db4",  
            "title": "hirings_workspace",  
            "updated": "2025-08-25T10:18:42.000+00:00",  
            "published": "2025-08-25T10:04:09.000+00:00",  
        }  
    ]  
}
```

```

        "links": [
            {
                "rel": "edit",
                "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
            }
        ],
        "content": {
            "name": "bws-type-definition",
            "properties": {
                "r_object_id": "080003e980007db4",
                "object_name": "hiring_workspace",
                "icon_name": "mime_wsjob_opening32",
                "locales": [
                    "en"
                ],
                "name_labels": [
                    "Buchhaltung case",
                    "Hiring Workspace"
                ],
                "descriptions": [
                    "Dies ist eine Beschreibung für die Buchhaltung",
                    "This is a description for hiring"
                ],
                "r_creator_name": "dmadmin",
                "r_creation_date": "2025-08-25T10:04:09.000+00:00",
                "r_modifier": "dmadmin",
                "r_modify_date": "2025-08-25T10:18:42.000+00:00"
            },
            "links": [
                {
                    "rel": "self",
                    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
                },
                {
                    "rel": "edit",
                    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
                },
                {
                    "rel": "http://identifiers.emc.com/linkrel/delete",
                    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
                },
                {
                    "rel": "http://identifiers.emc.com/linkrel/bws-folder-
templates",
                    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates"
                }
            ]
        }
    }
}

```



## 2.93.4 Get Workspace Icon Names

Retrieves all Workspace Icons.

### 2.93.4.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition?inline=true&links=false&view=icon\_name,object\_name

### 2.93.4.2 HTTP Request method

GET

### 2.93.4.3 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.93.4.4 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.93.4.5 Request body

N/A

### 2.93.4.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.93.4.7 Response media types

- application/vnd.emc.documentum+json
- application/json

#### 2.93.4.8 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.93.4.9 Response body

JSON representation.

The response is retrieved from the `dm_bws_type_def` type. Retrieves icons from all Workspaces. The response is optimized for the list view and includes only the necessary data required for display.

This involves a direct database call and therefore access validation is enforced. Data retrieval is permitted only if the logged in user is a superuser or is part of the `dm_bws_admin` role. For all other users, access is allowed only if the requested attributes are strictly limited to `icon_name` and `object_name`.

##### ➡ Example 2-470: JSON response

```
{
  "id": "/repositories/d2repo/bws-types-definition",
  "title": "List of Business Workspaces",
  "author": [
    {
      "name": "Open Text Documentum"
    }
  ],
  "updated": "2025-07-23T08:51:20.850+00:00",
  "page": 1,
  "items-per-page": 100,
  "links": [
    {
      "rel": "self",
      "href": "/repositories/d2repo/bws-types-definition?
      inline=true&links=false&view=icon_name,object_name"
    }
  ],
  "entries": [
    {
      "id": "/repositories/d2repo/bws-types-definition/080003e980005994",
      "title": "Legal132",
      "updated": "2025-07-23T08:51:20.850+00:00",
      "published": "2025-07-23T08:51:20.850+00:00",
      "links": [
        {
          "rel": "edit",
          "href": "/repositories/d2repo/bws-types-definition/080003e980005994/edit"
        }
      ]
    }
  ]
}
```

```
        "href": "/repositories/d2repo/bws-types-definition/
080003e980005994"
    }
],
"content": {
    "name": "bws-type-definition",
    "properties": {
        "icon_name": "hiring-workspace-icon",
        "object_name": "Legal132"
    }
}
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e9800059aa",
    "title": "Legal132",
    "updated": "2025-07-23T08:51:20.850+00:00",
    "published": "2025-07-23T08:51:20.850+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e9800059aa"
        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "hiring-workspace-icon",
            "object_name": "Legal132"
        }
    }
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e98000592e",
    "title": "PT",
    "updated": "2025-07-23T08:51:20.850+00:00",
    "published": "2025-07-23T08:51:20.850+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e98000592e"
        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "admin-workspace-icon",
            "object_name": "PT"
        }
    }
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e980005dcb",
    "title": "TEST_UU",
    "updated": "2025-07-23T08:51:20.850+00:00",
    "published": "2025-07-23T08:51:20.850+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980005dcb"
        }
    ]
}
```

```

        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "admin-workspace-icon",
            "object_name": "TEST_UU"
        }
    }
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e980005dbf",
    "title": "Test_UD",
    "updated": "2025-07-23T08:51:20.850+00:00",
    "published": "2025-07-23T08:51:20.850+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980005dbf"
        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "admin-workspace-icon",
            "object_name": "Test_UD"
        }
    }
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e98000592d",
    "title": "a1",
    "updated": "2025-07-23T08:51:20.850+00:00",
    "published": "2025-07-23T08:51:20.850+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e98000592d"
        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "admin-workspace-icon",
            "object_name": "a1"
        }
    }
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e980005d3f",
    "title": "code",
    "updated": "2025-07-23T08:51:20.850+00:00",
    "published": "2025-07-23T08:51:20.850+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980005d3f"
        }
    ],
}
]
,
```

```
        "content": {
            "name": "bws-type-definition",
            "properties": {
                "icon_name": "mime_wsteam32",
                "object_name": "code"
            }
        }
    },
    {
        "id": "/repositories/d2repo/bws-types-definition/080003e98000593e",
        "title": "hiring_workspace",
        "updated": "2025-07-23T08:51:20.850+00:00",
        "published": "2025-07-23T08:51:20.850+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "/repositories/d2repo/bws-types-definition/
080003e98000593e"
            }
        ],
        "content": {
            "name": "bws-type-definition",
            "properties": {
                "icon_name": "hiring-workspace-icon",
                "object_name": "hiring_workspace"
            }
        }
    },
    {
        "id": "/repositories/d2repo/bws-types-definition/080003e98000592f",
        "title": "name_label",
        "updated": "2025-07-23T08:51:20.850+00:00",
        "published": "2025-07-23T08:51:20.850+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "/repositories/d2repo/bws-types-definition/
080003e98000592f"
            }
        ],
        "content": {
            "name": "bws-type-definition",
            "properties": {
                "icon_name": "admin-workspace-icon",
                "object_name": "name_label"
            }
        }
    },
    {
        "id": "/repositories/d2repo/bws-types-definition/080003e98000592c",
        "title": "test",
        "updated": "2025-07-23T08:51:20.850+00:00",
        "published": "2025-07-23T08:51:20.850+00:00",
        "links": [
            {
                "rel": "edit",
                "href": "/repositories/d2repo/bws-types-definition/
080003e98000592c"
            }
        ],
        "content": {
            "name": "bws-type-definition",

```

```
        "properties": {
            "icon_name": "admin-workspace-icon",
            "object_name": "test"
        }
    },
{
    "id": "/repositories/d2repo/bws-types-definition/080003e980005d43",
    "title": "ud_test",
    "updated": "2025-07-23T08:51:20.851+00:00",
    "published": "2025-07-23T08:51:20.851+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980005d43"
        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "admin-workspace-icon",
            "object_name": "ud_test"
        }
    }
},
{
    "id": "/repositories/d2repo/bws-types-definition/080003e980005d91",
    "title": "vinay1",
    "updated": "2025-07-23T08:51:20.851+00:00",
    "published": "2025-07-23T08:51:20.851+00:00",
    "links": [
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980005d91"
        }
    ],
    "content": {
        "name": "bws-type-definition",
        "properties": {
            "icon_name": "hiring-workspace-icon",
            "object_name": "vinay1"
        }
    }
}
]
```



## 2.93.5 Update Workspace Type Definition

## **Updates a Workspace Type Definition.**

## 2.93.5.1 URI

- Item: Base URI
  - Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{r\_object\_id}?validate\_templates\_on\_acl\_update=true

### **2.93.5.2 HTTP Request method**

PATCH

### **2.93.5.3 Request media types**

- application/vnd.emc.documentum+json
  - application/json

## 2.93.5.4 Request headers

- Accept
  - Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### **2.93.5.5 Request body**

JSON representation.

The JSON request is stored in the database as content in the `dm_bws_type_def` type.

## → Example 2-471: JSON request

```
{  
    "permission-templates": [  
        {  
            "name": "Hiring PT",  
            "locales": [  
                "en",  
                "de"  
            ],  
            "name-labels": [  
                "Hiring PT",  
                "Einstellung von Teilzeitkräften"  
            ],  
            "descriptions": [  
                "This is a description for Hiring permission template.",  
                "Dies ist eine Beschreibung für die Vorlage zur  
Einstellgenehmigung."  
            ]  
        }  
    ]  
}
```

```

    "permitted": [
        {
            "accessor": "110003e980001548",
            "basic-permission": "Write",
            "accessor-type": "USER"
        },
        {
            "accessor": "120003e980000d01",
            "basic-permission": "Delete",
            "accessor-type": "GROUP"
        },
        {
            "accessor": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
            "basic-permission": "Delete",
            "accessor-type": "BWS"
        },
        {
            "accessor": "808cad06-60ee-47bb-96db-dfd166f4cb83",
            "basic-permission": "Read",
            "accessor-type": "BWS"
        },
        {
            "accessor": "dm_owner",
            "basic-permission": "Delete",
            "accessor-type": "ALIAS"
        },
        {
            "accessor": "dm_world",
            "basic-permission": "Read",
            "accessor-type": "ALIAS"
        }
    ]
}

```



### 2.93.5.6 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.93.5.7 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.93.5.8 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.93.5.9 Response body

JSON representation.

#### ➡ Example 2-472: JSON response sample

In the value of Base URI, if validate\_templates\_on\_acl\_update is set to false or not provided, and the updated permission template is already in use within Folder Templates, the system will move all associated Folder Templates to the Draft state.

```
{  
    "name": "bws-type-definition",  
    "type": "dm_bws_type_def",  
    "properties": {  
        "r_object_id": "080003e980007db4",  
        "object_name": "hiring_workspace",  
        "icon_name": "mime_wsjob_opening32",  
        "locales": [  
            "en"  
        ],  
        "name_labels": [  
            "Hiring Workspace",  
            "Buchhaltung case"  
        ],  
        "descriptions": [  
            "This is a description for hiring",  
            "Dies ist eine Beschreibung für die Buchhaltung"  
        ],  
        "r_creator_name": "dmadmin",  
        "r_creation_date": "2025-08-25T10:04:09.000+00:00",  
        "r_modifier": "dmadmin",  
        "r_modify_date": "2025-08-25T10:18:42.000+00:00"  
    },  
    "bws-folder-locations": [  
        {  
            "location-type": "id-location",  
            "id": "0b0003e98000750c",  
            "descendent": true  
        }  
    ],  
    "type-attributes": [  
        {  
            "labels": [  
                "is Admin",  
                "ist Admin"  
            ],  
            "locales": [  
                "en",  
                "de"  
            ]  
        },  
        {  
            "label": "is Admin",  
            "value": "true",  
            "type": "checkbox"  
        },  
        {  
            "label": "ist Admin",  
            "value": "true",  
            "type": "checkbox"  
        }  
    ]  
}
```

```
        "name": "is_admin",
        "repeating": false,
        "type": "boolean",
        "length": 0
    },
    {
        "labels": [
            "Account Number",
            "Kontonummer"
        ],
        "locales": [
            "en",
            "de"
        ],
        "name": "account_number",
        "repeating": false,
        "type": "integer",
        "length": 0
    },
    {
        "labels": [
            "Reference Id",
            "Referenz-ID"
        ],
        "locales": [
            "en",
            "de"
        ],
        "name": "reference_id",
        "repeating": false,
        "type": "id",
        "length": 0
    },
    {
        "labels": [
            "Followup Date",
            "Folgedatum"
        ],
        "locales": [
            "en",
            "de"
        ],
        "name": "follow_up_date",
        "repeating": false,
        "type": "time",
        "length": 0
    },
    {
        "labels": [
            "Version Number",
            "Versionsnummer"
        ],
        "locales": [
            "en",
            "de"
        ],
        "name": "version_number",
        "repeating": false,
        "type": "double",
        "length": 0
    }
],
```

```
"groups": [
    {
        "item-id": "808cad06-60ee-47bb-96db-dfd166f4cb83",
        "group-name": "Applicant",
        "locales": [
            "en"
        ],
        "descriptions": [
            "This is an applicant role",
            "Das ist eine Bewerberrolle"
        ],
        "group-labels": [
            "Applicant",
            "Applicante"
        ],
        "is-admin-role": false
    },
    {
        "item-id": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
        "group-name": "Hiring Manager",
        "locales": [
            "en"
        ],
        "descriptions": [
            "This is a description for Hiring Manager",
            "Dies ist eine Beschreibung für den Einstellungsleiter."
        ],
        "group-labels": [
            "Hiring Manager",
            "Einstellungsmanager"
        ],
        "is-admin-role": true
    }
],
"permission-templates": [
    {
        "item-id": "f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec",
        "name": "Hiring PT",
        "locales": [
            "en",
            "de"
        ],
        "name-labels": [
            "Hiring PT",
            "Einstellung von Teilzeitkräften"
        ],
        "descriptions": [
            "This is a description for Hiring permission template.",
            "Dies ist eine Beschreibung für die Vorlage zur Einstellgenehmigung."
        ],
        "permitted": [
            {
                "accessor-type": "USER",
                "accessor-name": "test_user_1",
                "accessor": "110003e980001548",
                "basic-permission": "Write"
            },
            {
                "accessor-type": "GROUP",
                "accessor-name": "admingroup",
                "accessor": "120003e980000d01",
                "basic-permission": "Read"
            }
        ]
    }
]
```

```

        "basic-permission": "Delete"
    },
    {
        "accessor-type": "BWS",
        "accessor": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "BWS",
        "accessor": "808cad06-60ee-47bb-96db-dfd166f4cb83",
        "basic-permission": "Read"
    },
    {
        "accessor-type": "ALIAS",
        "accessor": "dm_owner",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "ALIAS",
        "accessor": "dm_world",
        "basic-permission": "Read"
    }
]
},
],
"instance-exists": false,
"subtype-exists": false,
"links": [
{
    "rel": "self",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
},
{
    "rel": "edit",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
},
{
    "rel": "http://identifiers.emc.com/linkrel/delete",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
},
{
    "rel": "http://identifiers.emc.com/linkrel/bws-folder-templates",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates"
}
]
}
}

```



#### ➡ Example 2-473: JSON response sample for update conflict

In the value of Base URI, if validate\_templates\_on\_acl\_update is set to true and the updated permission template is already in use within Published Folder Templates, the system will return a 409 Conflict response along with the list of Folder Templates where the Permission Template is applied.

```
{  
    "status": 409,  
    "code": "E_BWS_CONFIRM_PT_UPDATE",  
    "message": "The modified Permission Template(s) is being referenced in  
following Folder Templates. These Folder Templates will be moved to Draft  
state. Do you want to continue? \nhiring ft",  
    "id": "fdf3a43f-9dd5-461c-a452-f1d738561341"  
}
```



#### Example 2-474: JSON response sample for delete conflict

If a Permission Template that is referenced in Folder Templates is deleted, the system will prevent the deletion and return a 409 Conflict response.

```
{  
    "status": 409,  
    "code": "E_BWS_ERROR_PT_DELETE",  
    "message": "Permission Template(s) cannot be deleted as it is being  
referenced in following Folder Templates: \nhiring ft",  
    "id": "69e3cd86-887a-4c32-b4f0-aaa7f799378a"  
}
```



### 2.93.6 Delete Workspace Type Definition

Deletes a Workspace Type Definition.

Users with the `dm_bws_admin` and `dm_bws_admin_dynamic` group or role can delete a type.

#### 2.93.6.1 URI

- Item: Base URI
- Value: `{repositoryUri}/repositories/{repositoryName}/bws-types-definition/{r_object_id}`

#### 2.93.6.2 HTTP Request method

DELETE

### 2.93.6.3 Request media types

N/A

### 2.93.6.4 Request headers

N/A

### 2.93.6.5 Request body

N/A

### 2.93.6.6 Response headers

N/A

### 2.93.6.7 Response media types

N/A

### 2.93.6.8 Response status

204 - No content

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.93.6.9 Response body

204 - No Content

## 2.94 Workspace Folder Template

### 2.94.1 Create Workspace Folder Template

Creates a Workspace Folder Template for a specific Workspace Type Definition.

#### 2.94.1.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{bwsTypeDefinitionId}/folder-templates

### 2.94.1.2 Navigation

Workspace Folder Template listing page

### 2.94.1.3 Link relations

<http://identifiers.emc.com/linkrel/bws-folder-templates>

### 2.94.1.4 HTTP Request method

POST

### 2.94.1.5 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.1.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.94.1.7 Request body

The JSON request is stored in the database as content in the dm\_bws\_template type.

The items specified in the properties section in the example request body sample are stored as columns and the remaining are stored as JSON content.

#### ➡ Example 2-475: JSON request

```
{  
    "properties": {  
        "object_name": "Software Developer Hiring",  
        "locales": [  
            "en",  
            "de"  
        ],  
        "name_labels": [  
            "Software Developer Hiring",  
            "Einstellung von Softwareentwicklern"  
        ],  
        "descriptions": [  
            "This template is for hiring software developer.",  
            "Diese Vorlage ist für die Einstellung von Softwareentwicklern."  
        ]  
    },  
    "case-model": {  
        "root-case-node": {  
            "name": "Documents",  
            "label": "Software Development Hiring",  
            "description": "This template is for hiring software developer.",  
            "parent": null,  
            "children": []  
        }  
    }  
}
```

```

    "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-
cb8f3e1fafec",
    "child-nodes": [
        {
            "name": "Academic Documents",
            "doc-types": [
                {
                    "name": "c2_file",
                    "permission-template-item-id":
"f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec"
                }
            ],
            "child-nodes": [
                {
                    "name": "Projects"
                }
            ]
        },
        {
            "name": "Employment Documents",
            "doc-types": [
                {
                    "name": "experience_letter"
                }
            ]
        }
    ]
}

```



#### 2.94.1.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.94.1.9 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.1.10 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.94.1.11 Response body

JSON representation.

#### ➡ Example 2-476: JSON response

```
{  
    "name": "bws-folder-template",  
    "type": "dm_bws_template",  
    "properties": {  
        "r_object_id": "080003e980007db6",  
        "object_name": "Software Developer Hiring",  
        "locales": [  
            "en",  
            "de"  
        ],  
        "name_labels": [  
            "Software Developer Hiring",  
            "Einstellung von Softwareentwicklern"  
        ],  
        "descriptions": [  
            "This template is for hiring software developer.",  
            "Diese Vorlage ist für die Einstellung von Softwareentwicklern."  
        ],  
        "bws_type_def_id": "080003e980007db4",  
        "r_version_label": [  
            "1.0",  
            "CURRENT"  
        ],  
        "r_creator_name": "dmadmin",  
        "r_creation_date": "2025-08-25T10:59:29.000+00:00",  
        "r_modifier": "dmadmin",  
        "r_modify_date": "2025-08-25T10:59:29.000+00:00",  
        "template_state": "DRAFT"  
    },  
    "case-model": {  
        "root-case-node": {  
            "name": "Documents",  
            "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-  
cb8f3e1fafec",  
            "child-nodes": [  
                {  
                    "case-node-id": "2bf17946-40f1-4398-834b-6e27c148cd1c",  
                    "name": "Academic Documents",  
                    "child-nodes": [  
                        {  
                            "case-node-id":  
                            "a2080cf4-82ed-41c5-815a-7cd28defde5e",  
                            "name": "Projects"  
                        }  
                    ]  
                },  
                "doc-types": [  
                ]  
            ]  
        }  
    }  
}
```

```
{  
    "name": "c2_file",  
    "permission-template-item-id":  
"f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec"  
}  
]  
}  
{  
    "case-node-id": "ddef01c9-af61-406c-b1a9-296f7f33b063",  
    "name": "Employment Documents",  
    "doc-types": [  
        {  
            "name": "experience_letter"  
        }  
    ]  
},  
    "case-node-id": "735b4ead-cdd5-4c30-b1a8-fe7c2862efdf"  
}  
},  
    "instance-exists": false,  
    "links": [  
        {  
            "rel": "self",  
            "href": "/repositories/d2repo/bws-types-definition/  
080003e980007db4/folder-templates/080003e980007db6"  
        },  
        {  
            "rel": "edit",  
            "href": "/repositories/d2repo/bws-types-definition/  
080003e980007db4/folder-templates/080003e980007db6"  
        },  
        {  
            "rel": "http://identifiers.emc.com/linkrel/delete",  
            "href": "/repositories/d2repo/bws-types-definition/  
080003e980007db4/folder-templates/080003e980007db6"  
        },  
        {  
            "rel": "parent",  
            "href": "/repositories/d2repo/bws-types-definition/  
080003e980007db4"  
        }  
    ]  
}
```



## 2.94.2 Get Workspace Folder Template

Retrieves an individual Workspace Folder Template for a specific Workspace Type Definition.

If Workspace Folder Template is in published state, the associated groups and permission template information is retrieved.

### 2.94.2.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{bwsTypeDefinitionId}/folder-templates/{bwsFolderTemplateId}

### 2.94.2.2 Navigation

From the list of Workspace Folder Templates available in the Workspace Type page

### 2.94.2.3 Link relations

self

### 2.94.2.4 HTTP Request method

GET

### 2.94.2.5 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.2.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.94.2.7 Request body

N/A

### 2.94.2.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.94.2.9 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.2.10 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.94.2.11 Response body

JSON representation.

The response is retrieved from the `d2_bws_template` type. The response is extracted from the content. All the required Workspace information is available in the response.

#### ➡ Example 2-477: JSON response

```
{
    "name": "bws-folder-template",
    "type": "dm_bws_template",
    "properties": {
        "r_object_id": "080003e980007db6",
        "object_name": "Software Developer Hiring",
        "locales": [
            "en",
            "de"
        ],
        "name_labels": [
            "Software Developer Hiring",
            "Einstellung von Softwareentwicklern"
        ],
        "descriptions": [
            "This template is for hiring software developer.",
            "Diese Vorlage ist für die Einstellung von Softwareentwicklern."
        ],
        "bws_type_def_id": "080003e980007db4",
        "r_version_label": [

```

```
        "1.0"
    ],
    "r_creator_name": "dmadmin",
    "r_creation_date": "2025-08-25T10:59:29.000+00:00",
    "r_modifier": "dmadmin",
    "r_modify_date": "2025-08-25T11:13:34.000+00:00",
    "template_state": "PUBLISHED"
},
"groups": [
{
    "item-id": "808cad06-60ee-47bb-96db-dfd166f4cb83",
    "group-name": "Applicant",
    "locales": [
        "en"
    ],
    "descriptions": [
        "This is an applicant role",
        "Das ist eine Bewerberrolle"
    ],
    "group-labels": [
        "Applicant",
        "Applicante"
    ],
    "is-admin-role": false
},
{
    "item-id": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
    "group-name": "Hiring Manager",
    "locales": [
        "en"
    ],
    "descriptions": [
        "This is a description for Hiring Manager",
        "Dies ist eine Beschreibung für den Einstellungsleiter."
    ],
    "group-labels": [
        "Hiring Manager",
        "Einstellungsmanager"
    ],
    "is-admin-role": true
}
],
"permission-templates": [
{
    "item-id": "f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec",
    "name": "Hiring PT",
    "locales": [
        "en",
        "de"
    ],
    "name-labels": [
        "Hiring PT",
        "Einstellung von Teilzeitkräften"
    ],
    "descriptions": [
        "This is a description for Hiring permission template.",
        "Dies ist eine Beschreibung für die Vorlage zur Einstellgenehmigung."
    ],
    "permitted": [
        {
            "accessor-type": "USER",

```

```

        "accessor": "110003e980001548",
        "basic-permission": "Write"
    },
    {
        "accessor-type": "GROUP",
        "accessor": "120003e980000d01",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "BWS",
        "accessor": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "BWS",
        "accessor": "808cad06-60ee-47bb-96db-dfd166f4cb83",
        "basic-permission": "Read"
    },
    {
        "accessor-type": "ALIAS",
        "accessor": "dm_owner",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "ALIAS",
        "accessor": "dm_world",
        "basic-permission": "Read"
    }
]
},
],
"case-model": {
    "root-case-node": {
        "name": "Documents",
        "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-
cb8f3e1fafec",
        "child-nodes": [
            {
                "case-node-id": "2bf17946-40f1-4398-834b-6e27c148cd1c",
                "name": "Academic Documents",
                "child-nodes": [
                    {
                        "case-node-id":
                            "a2080cf4-82ed-41c5-815a-7cd28defde5e",
                        "name": "Projects"
                    }
                ],
                "doc-types": [
                    {
                        "name": "c2_file",
                        "permission-template-item-id":
                            "f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec"
                    }
                ]
            },
            {
                "case-node-id": "ddef01c9-af61-406c-b1a9-296f7f33b063",
                "name": "Employment Documents",
                "doc-types": [
                    {
                        "name": "experience_letter"
                    }
                ]
            }
        ]
    }
}

```

```
        ]
    }
],
"case-node-id": "735b4ead-cdd5-4c30-b1a8-fe7c2862efdf"
}
},
"instance-exists": false,
"links": [
{
    "rel": "self",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007db6"
},
{
    "rel": "edit",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007db6"
},
{
    "rel": "http://identifiers.emc.com/linkrel/delete",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007db6"
},
{
    "rel": "parent",
    "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
}
]
}
```



### 2.94.3 Get All Workspace Folder Template

Retrieves all Workspace Folder Templates associated with a specific Workspace Type Definition.

#### 2.94.3.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{bwsTypeDefinitionId}/folder-templates

### 2.94.3.2 Navigation

Workspace Type Definition page

### 2.94.3.3 Link relations

<http://identifiers.emc.com/linkrel/bws-folder-templates>

### 2.94.3.4 HTTP Request method

GET

### 2.94.3.5 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.3.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.94.3.7 Request body

N/A

### 2.94.3.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.94.3.9 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.3.10 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.94.3.11 Response body

JSON representation.

The response is optimized for the list view and includes only the necessary data required for display. All the Workspace Folder Template information is stored as JSON content and excluded in the response.

#### ➡ Example 2-478: JSON response

```
{  
    "id": "/repositories/d2repo/bws-types-definition/080003e980007db4/folder-  
    templates",  
    "title": "List of Folder Templates",  
    "author": [  
        {  
            "name": "Open Text Documentum"  
        }  
    ],  
    "updated": "2025-08-25T11:18:20.484+00:00",  
    "page": 1,  
    "items-per-page": 100,  
    "links": [  
        {  
            "rel": "self",  
            "href": "/repositories/d2repo/bws-types-definition/  
080003e980007db4/folder-templates?inline=true&links=true"  
        }  
    ],  
    "entries": [  
        {  
            "id": "/repositories/d2repo/bws-types-definition/080003e980007db4/  
folder-templates/080003e980007dd0",  
            "title": "Software Developer Hiring",  
            "updated": "2025-08-25T11:13:34.000+00:00",  
            "published": "2025-08-25T11:13:34.000+00:00",  
            "links": [  
                {  
                    "rel": "edit",  
                    "href": "/repositories/d2repo/bws-types-definition/  
080003e980007db4/folder-templates/080003e980007dd0"  
                }  
            ],  
            "content": {  
                "name": "bws-folder-template",  
                "properties": {  
                    "r_object_id": "080003e980007dd0",  
                    "object_name": "Software Developer Hiring",  
                    "locales": [  
                        "de",  
                        "en"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```

        ],
        "name_labels": [
            "Einstellung von Softwareentwicklern",
            "Software Developer Hiring"
        ],
        "descriptions": [
            "Diese Vorlage ist für die Einstellung von
Softwareentwicklern.", "This template is for hiring software developer."
        ],
        "bws_type_def_id": "080003e980007db4",
        "r_version_label": [
            "CURRENT",
            "1.1"
        ],
        "r_creator_name": "dmadmin",
        "r_creation_date": "2025-08-25T11:13:34.000+00:00",
        "r_modifier": "dmadmin",
        "r_modify_date": "2025-08-25T11:13:34.000+00:00",
        "template_state": "DRAFT"
    },
    "links": [
        {
            "rel": "self",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007dd0"
        },
        {
            "rel": "edit",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007dd0"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/delete",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007dd0"
        },
        {
            "rel": "parent",
            "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
        }
    ]
}
}

```



## 2.94.4 Publish Workspace Folder Template

Publishes the Workspace Folder Template. To change the state of the Workspace Folder Template, provide the value of template\_state to 1 as follows:

```
template_state=1
```

To publish a Workspace Folder Template, provide the value of the action request parameter to PUBLISH as follows:

```
action=PUBLISH
```

### 2.94.4.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{bwsTypeDefinitionId}/folder-templates/{bwsFolderTemplateId}?action=PUBLISH

### 2.94.4.2 Navigation

From the list of Workspace Folder Templates available in the Workspace Type Definition page

### 2.94.4.3 Link relations

self

### 2.94.4.4 HTTP Request method

PATCH

### 2.94.4.5 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.94.4.6 Request headers

- Accept
- Authorization

For more information, see Appendix D, REST Common Definition - HTTP Headers on page 1055.

#### 2.94.4.7 Request body

```
1 {}
```

#### 2.94.4.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.94.4.9 Response media types

- application/vnd.emc.documentum+json
- application/json

#### 2.94.4.10 Response status

- 200 - Published successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### 2.94.4.11 Response body

JSON representation.

The request body for a patch should not empty.

##### Example 2-479: JSON response

```
{
    "name": "bws-folder-template",
    "type": "dm_bws_template",
    "properties": {
        "r_object_id": "080003e980007db6",
        "object_name": "Software Developer Hiring",
        "locales": [
            "en",
            "de"
        ],
        "name_labels": [
            "Software Developer Hiring",
            "Einstellung von Softwareentwicklern"
        ],
        "descriptions": [
            "This template is for hiring software developer.",
            "Diese Vorlage ist für die Einstellung von Softwareentwicklern."
        ],
        "bws_type_def_id": "080003e980007db4",
        "r_version_label": [
            "1.0",
            "CURRENT"
        ]
    }
}
```

```
        ],
        "r_creator_name": "dmadmin",
        "r_creation_date": "2025-08-25T10:59:29.000+00:00",
        "r_modifier": "dmadmin",
        "r_modify_date": "2025-08-25T11:02:17.000+00:00",
        "template_state": "PUBLISHED"
    },
    "groups": [
        {
            "item-id": "808cad06-60ee-47bb-96db-dfd166f4cb83",
            "group-name": "Applicant",
            "locales": [
                "en"
            ],
            "descriptions": [
                "This is an applicant role",
                "Das ist eine Bewerberrolle"
            ],
            "group-labels": [
                "Applicant",
                "Applicante"
            ],
            "is-admin-role": false
        },
        {
            "item-id": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
            "group-name": "Hiring Manager",
            "locales": [
                "en"
            ],
            "descriptions": [
                "This is a description for Hiring Manager",
                "Dies ist eine Beschreibung für den Einstellungsleiter."
            ],
            "group-labels": [
                "Hiring Manager",
                "Einstellungsmanager"
            ],
            "is-admin-role": true
        }
    ],
    "permission-templates": [
        {
            "item-id": "f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec",
            "name": "Hiring PT",
            "locales": [
                "en",
                "de"
            ],
            "name-labels": [
                "Hiring PT",
                "Einstellung von Teilzeitkräften"
            ],
            "descriptions": [
                "This is a description for Hiring permission template.",
                "Dies ist eine Beschreibung für die Vorlage zur Einstellgenehmigung."
            ],
            "permitted": [
                {
                    "accessor-type": "USER",
                    "accessor": "110003e980001548",
                    "actions": [
                        "CREATE"
                    ]
                }
            ]
        }
    ]
}
```

```

        "basic-permission": "Write"
    },
    {
        "accessor-type": "GROUP",
        "accessor": "120003e980000d01",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "BWS",
        "accessor": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "BWS",
        "accessor": "808cad06-60ee-47bb-96db-dfd166f4cb83",
        "basic-permission": "Read"
    },
    {
        "accessor-type": "ALIAS",
        "accessor": "dm_owner",
        "basic-permission": "Delete"
    },
    {
        "accessor-type": "ALIAS",
        "accessor": "dm_world",
        "basic-permission": "Read"
    }
]
},
{
    "case-model": {
        "root-case-node": {
            "name": "Documents",
            "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-
cb8f3e1fafec",
            "child-nodes": [
                {
                    "case-node-id": "b7aebaa1-c401-43fd-a1d9-336872de2d53",
                    "name": "Academic Documents",
                    "doc-types": [
                        {
                            "name": "c2_file"
                        }
                    ]
                },
                {
                    "case-node-id": "8cf49999-32e4-47ee-a1f3-0226424497cf",
                    "name": "Employment Documents",
                    "doc-types": [
                        {
                            "name": "c2_file"
                        }
                    ]
                }
            ],
            "case-node-id": "cf79be31-3bd7-4c38-a4b8-0e81f160892b"
        }
    },
    "instance-exists": false,
    "links": [
        {
            "rel": "self",

```

```
        "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007db6"
    },
    {
        "rel": "edit",
        "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007db6"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007db6"
    },
    {
        "rel": "parent",
        "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
    }
]
```



## 2.94.5 Update Workspace Folder Template

Updates a Workspace Folder Template.

### 2.94.5.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/bws-types-definition/{bwsTypeDefinitionId}/folder-templates/{bwsFolderTemplateId}

### 2.94.5.2 Navigation

From the list of Workspace Folder Templates available in the Workspace Type Definition page

### 2.94.5.3 Link relations

self

#### 2.94.5.4 HTTP Request method

PATCH

#### 2.94.5.5 Request media types

- application/vnd.emc.documentum+json
- application/json

#### 2.94.5.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.94.5.7 Request body

The request body is submitted either as individual sections or as a complete payload. The properties section allows partial updates and only the modified properties need to be included. However, for the case\_model section, if any property is updated, the entire case\_model object must be included in the request. While the request body supports partial updates by section, the response body always returns the complete and updated Workspace Folder Template.

 **Example 2-480: JSON request sample 1**

```
{
  "properties": {
    "object_name": "Software Developer Hiring",
    "locales": [
      "en",
      "de"
    ],
    "name_labels": [
      "Software Developer Hiring",
      "Einstellung von Softwareentwicklern"
    ],
    "descriptions": [
      "This template is for hiring software developer.",
      "Diese Vorlage ist für die Einstellung von Softwareentwicklern."
    ]
  }
}
```



 **Example 2-481: JSON request sample 2**

```
{
  "case-model": {
    "root-case-node": {
      "name": "Documents",
      "children": [
        ...
      ]
    }
  }
}
```

```
        "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-  
cb8f3e1fafec",  
        "child-nodes": [  
            {  
                "name": "Academic Documents",  
                "doc-types": [  
                    {  
                        "name": "c2_file"  
                    }  
                ]  
            },  
            {  
                "name": "Employment Documents",  
                "doc-types": [  
                    {  
                        "name": "c2_file"  
                    }  
                ]  
            },  
            {  
                "name": "Employer Documents",  
                "doc-types": [  
                    {  
                        "name": "c2_file"  
                    }  
                ]  
            }  
        ]  
    }  
}
```



## Example 2-482: JSON request sample 3

```
{
  "properties": {
    "object_name": "Software Developer Hiring",
    "locales": [
      "en",
      "de"
    ],
    "name_labels": [
      "Software Developer Hiring",
      "Einstellung von Softwareentwicklern"
    ],
    "descriptions": [
      "This template is for hiring software developer.",
      "Diese Vorlage ist für die Einstellung von Softwareentwicklern."
    ]
  },
  "case-model": {
    "root-case-node": {
      "name": "Documents",
      "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-
cb8f3e1fafec",
      "child-nodes": [
        {
          "name": "Academic Documents",
          "doc-types": [
            "Academic Document"
          ]
        }
      ]
    }
  }
}
```

```

        {
          "name": "c2_file"
        }
      ],
    },
    {
      "name": "Employment Documents",
      "doc-types": [
        {
          "name": "c2_file"
        }
      ],
    },
    {
      "name": "Employer Documents",
      "doc-types": [
        {
          "name": "c2_file"
        }
      ],
    }
  ]
}

```



#### 2.94.5.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.94.5.9 Response media types

- application/vnd.emc.documentum+json
- application/json

#### 2.94.5.10 Response status

- 200 - Updated successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.94.5.11 Response body

JSON representation.

#### ➡ Example 2-483: JSON response

```
{
    "name": "bws-folder-template",
    "type": "dm_bws_template",
    "properties": {
        "r_object_id": "080003e980007dd0",
        "object_name": "Software Developer Hiring",
        "locales": [
            "en",
            "de"
        ],
        "name_labels": [
            "Software Developer Hiring",
            "Einstellung von Softwareentwicklern"
        ],
        "descriptions": [
            "This template is for hiring software developer.",
            "Diese Vorlage ist für die Einstellung von Softwareentwicklern."
        ],
        "bws_type_def_id": "080003e980007db4",
        "r_version_label": [
            "1.1",
            "CURRENT"
        ],
        "r_creator_name": "dmadmin",
        "r_creation_date": "2025-08-25T11:13:34.000+00:00",
        "r_modifier": "dmadmin",
        "r_modify_date": "2025-08-25T11:13:34.000+00:00",
        "template_state": "DRAFT"
    },
    "groups": [
        {
            "item-id": "808cad06-60ee-47bb-96db-dfd166f4cb83",
            "group-name": "Applicant",
            "locales": [
                "en"
            ],
            "descriptions": [
                "This is an applicant role",
                "Das ist eine Bewerberrolle"
            ],
            "group-labels": [
                "Applicant",
                "Applicante"
            ],
            "is-admin-role": false
        },
        {
            "item-id": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
            "group-name": "Hiring Manager",
            "locales": [
                "en"
            ],
            "descriptions": [
                "This is a description for Hiring Manager",
                "Dies ist eine Beschreibung für den Einstellungsleiter."
            ],
        }
    ]
}
```

```

        "group-labels": [
            "Hiring Manager",
            "Einstellungsmanager"
        ],
        "is-admin-role": true
    }
],
"permission-templates": [
{
    "item-id": "f3256d7b-5597-4a8e-9ebc-cb8f3e1fafec",
    "name": "Hiring PT",
    "locales": [
        "en",
        "de"
    ],
    "name-labels": [
        "Hiring PT",
        "Einstellung von Teilzeitkräften"
    ],
    "descriptions": [
        "This is a description for Hiring permission template.",
        "Dies ist eine Beschreibung für die Vorlage zur Einstellgenehmigung."
    ],
    "permitted": [
        {
            "accessor-type": "USER",
            "accessor": "110003e980001548",
            "basic-permission": "Write"
        },
        {
            "accessor-type": "GROUP",
            "accessor": "120003e980000d01",
            "basic-permission": "Delete"
        },
        {
            "accessor-type": "BWS",
            "accessor": "3f1e0fe8-c4bb-408e-9bce-344566d0105d",
            "basic-permission": "Delete"
        },
        {
            "accessor-type": "BWS",
            "accessor": "808cad06-60ee-47bb-96db-dfd166f4cb83",
            "basic-permission": "Read"
        },
        {
            "accessor-type": "ALIAS",
            "accessor": "dm_owner",
            "basic-permission": "Delete"
        },
        {
            "accessor-type": "ALIAS",
            "accessor": "dm_world",
            "basic-permission": "Read"
        }
    ]
}
],
"case-model": {
    "root-case-node": {
        "name": "Documents",
        "permission-template-item-id": "f3256d7b-5597-4a8e-9ebc-

```

```
cb8f3e1fafec",
  "child-nodes": [
    {
      "case-node-id": "a7e591c2-ca54-43da-a8dd-e62d9502ffb9",
      "name": "Academic Documents",
      "doc-types": [
        {
          "name": "c2_file"
        }
      ]
    },
    {
      "case-node-id": "9d2b2b68-1130-43fd-9894-de34a7256bfb",
      "name": "Employment Documents",
      "doc-types": [
        {
          "name": "c2_file"
        }
      ]
    },
    {
      "case-node-id": "9237ad02-921c-4837-94b2-1839571db0ef",
      "name": "Employer Documents",
      "doc-types": [
        {
          "name": "c2_file"
        }
      ]
    }
  ],
  "case-node-id": "64e9aa5c-ae71-4f50-b301-674a32aa099b"
},
{
  "instance-exists": false,
  "links": [
    {
      "rel": "self",
      "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007dd0"
    },
    {
      "rel": "edit",
      "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007dd0"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/delete",
      "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4/folder-templates/080003e980007dd0"
    },
    {
      "rel": "parent",
      "href": "/repositories/d2repo/bws-types-definition/
080003e980007db4"
    }
  ]
}
```



## 2.94.6 Delete Workspace Folder Template

Deletes an individual Workspace Folder Template for a specific Workspace Type Definition.

Users with the `dm_bws_admin` and `dm_bws_admin_dynamic` group or role can delete a Workspace Folder Template.

### 2.94.6.1 URI

- Item: Base URI
- Value: `{repositoryUri}/repositories/{repositoryName}/bws-types-definition/{r_object_id}`

### 2.94.6.2 Navigation

From the list of Workspace Folder Templates available in the Workspace Type Definition page

### 2.94.6.3 Link relations

self

### 2.94.6.4 HTTP Request method

DELETE

### 2.94.6.5 Request media types

- `application/vnd.emc.documentum+json`
- `application/json`

### 2.94.6.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### **2.94.6.7 Request body**

N/A

#### **2.94.6.8 Response headers**

N/A

#### **2.94.6.9 Response media types**

N/A

#### **2.94.6.10 Response status**

204 - No Content

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

#### **2.94.6.11 Response body**

204 - No Content

### **2.95 Workspace Instance**

#### **2.95.1 Create Workspace Instance**

Creates a Workspace Instance.

During the Workspace runtime, the Workspace related objects such as folder structure hierarchy, ACLs , user groups and roles, and so on are generated. Select a Workspace template for this API to use the template ID which has the blueprint for the runtime object creation. It also uses Foundation Java APIs to generate the objects.

##### **2.95.1.1 URI**

- Item: Base URI
- Value: /repositories/<repository-name>/business-workspaces

### 2.95.1.2 Navigation

Repository

### 2.95.1.3 Link relations

<http://identifiers.emc.com/linkrel/business-workspaces>

### 2.95.1.4 HTTP Request method

POST

### 2.95.1.5 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.95.1.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.95.1.7 Request body

► Example 2-484: JSON request with type

```
{
  "template-id": "080003e980005aa7",
  "bws-parent-folder-id": "0b0003e980005965",
  "type": "student",
  "properties": {
    "object_name": "Instance1",
    "firstname": "Jane2",
    "lastname": "Doe",
    "course": "MBBS"
  }
}
```



► Example 2-485: JSON request without type

```
{
  "template-id": "080003e980005aa7",
  "bws-parent-folder-id": "0b0003e980005965",
  "properties": {
    "object_name": "Instance1",
    "firstname": "Jane2",
    "lastname": "Doe",
    "course": "MBBS"
  }
}
```



- `template-id`: Represents the template to be used to create the instance. This contains the blueprint for the folder structure, roles, permission templates, and so on.
- `bws-parent-folder-id`: Represents the folder ID where you want to create the instance.
- `type`: Type definition name, essentially the type you want to create the instance. From the Smart View UI, the type definition corresponding to `template-id` information is made available to populate the type. However, from all other client applications, the type definition may not be available. In this scenario, do not provide a value for this field. Instead, a query is used to retrieve the type definition's object name for the specified `template-id`. If an invalid type is provided which is not inherited from the `dm_bws` type, it results in an error message indicating about the non-existent type.
- `properties`: Includes the property bag which should be saved for the instance. This can contain a mix of inherited and custom attributes.

#### 2.95.1.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

#### 2.95.1.9 Response media types

- `application/vnd.emc.documentum+json`
- `application/json`

#### 2.95.1.10 Response status

- 201 - Created successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.95.1.11 Response body

JSON representation.

#### ▶ Example 2-486: JSON response

```
{
    "name": "bws-object",
    "type": "student",
    "definition": "http://localhost:8183/documentum-rest-web/repositories/d2repo/types/student",
    "properties": {
        "object_name": "Instance1",
        "r_object_type": "student",
        "title": "",
        "subject": "",
        "authors": null,
        "keywords": null,
        "a_application_type": "",
        "a_status": "",
        "r_creation_date": "2025-08-29T09:42:56.000+00:00",
        "r_modify_date": "2025-08-29T09:42:56.000+00:00",
        "r_modifier": "dmadmin",
        "r_access_date": null,
        "a_is_hidden": false,
        "i_is_deleted": false,
        "a_retention_date": null,
        "a_archive": false,
        "a_compound_architecture": "",
        "a_link_resolved": false,
        "i_reference_cnt": 1,
        "i_has_folder": true,
        "i_folder_id": [
            "0b0003e980005965"
        ],
        "r_composite_id": null,
        "r_composite_label": null,
        "r_component_label": null,
        "r_order_no": null,
        "r_link_cnt": 1,
        "r_link_high_cnt": 0,
        "r_assembled_from_id": "00000000000000000000",
        "r_frnz_n_assembly_cnt": 0,
        "r_has_frnz_n_assembly": false,
        "resolution_label": "",
        "r_is_virtual_doc": 0,
        "i_contents_id": "0000000000000000",
        "a_content_type": "",
        "r_page_cnt": 0,
        "r_content_size": 0,
        "a_full_text": true,
        "a_storage_type": "",
        "i_cabinet_id": "0c0003e980005a61",
        "owner_name": "dmadmin",
        "owner_permit": 7,
        "group_name": "docu",
        "group_permit": 1,
        "world_permit": 3,
        "i_antecedent_id": "0000000000000000",
        "i_chronicle_id": "810003e98000210b",
        "i_latest_flag": true,
        "r_lock_owner": ""
    }
}
```

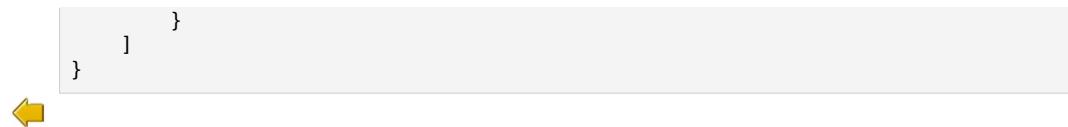
```
"r_lock_date": null,
"r_lock_machine": "",
"log_entry": "",
"r_version_label": [
    "1.0",
    "CURRENT"
],
"i_branch_cnt": 0,
"i_direct_dsc": false,
"r_immutable_flag": false,
"r_frozen_flag": false,
"r_has_events": false,
"acl_domain": "dmadmin",
"acl_name": "1_bws_810003e98000210b",
"a_special_app": "",
"i_is_reference": false,
"r_creator_name": "dmadmin",
"r_is_public": true,
"r_policy_id": "0000000000000000",
"r_resume_state": 0,
"r_current_state": 0,
"r_alias_set_id": "0000000000000000",
"a_effective_date": null,
"a_expiration_date": null,
"a_publish_formats": null,
"a_effective_label": null,
"a_effective_flag": null,
"a_category": "",
"language_code": "",
"a_is_template": false,
"a_controlling_app": "",
"r_full_content_size": 0.0,
"a_extended_properties": null,
"i_is_signed": false,
"i_last_review_date": null,
"i_retain_until": null,
"r_aspect_name": null,
"i_retainer_id": null,
"i_partition": 0,
"i_is_replica": false,
"i_vstamp": 1,
"r_folder_path": [
    "/DeeShan/Students/Instance3"
],
"i_ancestor_id": [
    "810003e98000210b",
    "0b0003e980005965",
    "0c0003e980005a61"
],
"template_id": "080003e980005aa7",
"firstname": "Jane2",
"lastname": "Doe",
"course": "MBBS",
"r_object_id": "810003e98000210b"
},
"links": [
{
    "rel": "self",
    "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/business-workspaces/810003e98000210b"
}
]
```

```

        "rel": "http://identifiers.emc.com/linkrel/parent-links",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/parent-links"
    },
    {
        "rel": "parent",
        "title": "0b0003e980005965",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/0b0003e980005965"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/cabinet",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/cabinets/0c0003e980005a61"
    },
    {
        "rel": "contents",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/contents"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/primary-content",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/contents/content"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/checkout",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/lock"
    },
    {
        "rel": "version-history",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/versions"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/current-version",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/versions/current"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/virtual-document-nodes",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/vd-nodes"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/snapshots",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/snapshots?filter=r_assembled_from_id%20%3D'810003e98000210b'"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/audit-trails",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/audit-trails?filter=audited_obj_id='810003e98000210b'"
    },
    {
        "rel": "edit",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",

```

```
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/folders",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/folders"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/documents",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/documents"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/objects",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/objects"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/child-links",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/child-links"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/relations",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/relations?related-object-id=810003e98000210b&related-object-role=any"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/object-aspects",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/aspects"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/permission-set",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/permission-set"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/permissions",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/permissions"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/comments",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/comments"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/attachable-lifecycles",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/lifecycles?object-id=810003e98000210b"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/object-lifecycle",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/lifecycle"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/bws-roles",
        "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/business-workspaces/810003e98000210b/roles"
```



## 2.95.2 Get a Workspace Instance

Retrieves a Workspace Instance.

### 2.95.2.1 URI

- Item: Base URI
- Value: {repositoryUri}/repositories/{repositoryName}/business-workspaces/{id}

### 2.95.2.2 Navigation

Workspaces

### 2.95.2.3 Link relations

self

### 2.95.2.4 HTTP Request method

GET

### 2.95.2.5 Request media types

- application/vnd.emc.documentum+json
- application/json

### 2.95.2.6 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.95.2.7 Request body

N/A

### 2.95.2.8 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers on page 1055](#).

### 2.95.2.9 Response media types

- application/vnd.emc.documentum+json
- application/json

### 2.95.2.10 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes on page 1053](#).

### 2.95.2.11 Response body

JSON representation.

The response includes the complete details of the object. In addition, the `bws_type_def_description` field provides the localized description of the workspace definition, and the `bws_type_def_label_name` field contains the localized label of the workspace definition. If a localized label is not available, the system sets the value to English. If an English label is also unavailable, the object name of the workspace definition is used as the fallback option.

#### ➡ Example 2-487: JSON response

```
{  
    "name": "bws-object",  
    "type": "student",  
    "definition": "http://localhost:8183/documentum-rest-web/repositories/  
d2repo/types/student",  
    "properties": {  
        "object_name": "Instance3",  
        "r_object_type": "student",  
        "r_creation_date": "2025-08-29T09:42:56.000+00:00",  
        "r_modify_date": "2025-08-29T09:42:56.000+00:00",  
        "r_modifier": "dmadmin",  
        "a_is_hidden": false,  
        "i_is_deleted": false,  
        "a_archive": false,  
        "a_link_resolved": false,  
        "i_reference_cnt": 1,  
    }  
}
```

```

    "i_has_folder": true,
    "i_folder_id": [
        "0b0003e980005965"
    ],
    "r_link_cnt": 3,
    "r_link_high_cnt": 0,
    "r_assembled_from_id": "0000000000000000",
    "r_frnz_n_assembly_cnt": 0,
    "r_has_frnz_n_assembly": false,
    "r_is_virtual_doc": 0,
    "i_contents_id": "0000000000000000",
    "r_page_cnt": 0,
    "r_content_size": 0,
    "a_full_text": true,
    "i_cabinet_id": "0c0003e980005a61",
    "owner_name": "dmadmin",
    "owner_permit": 7,
    "group_name": "docu",
    "group_permit": 1,
    "world_permit": 3,
    "i_antecedent_id": "0000000000000000",
    "i_chronicle_id": "810003e98000210b",
    "i_latest_flag": true,
    "r_version_label": [
        "1.0",
        "CURRENT"
    ],
    "i_branch_cnt": 0,
    "i_direct_dsc": false,
    "r_immutable_flag": false,
    "r_frozen_flag": false,
    "r_has_events": false,
    "acl_domain": "dmadmin",
    "acl_name": "1_bws_810003e98000210b",
    "i_is_reference": false,
    "r_creator_name": "dmadmin",
    "r_is_public": true,
    "r_policy_id": "0000000000000000",
    "r_resume_state": 0,
    "r_current_state": 0,
    "r_alias_set_id": "0000000000000000",
    "a_is_template": false,
    "r_full_content_size": 0.0,
    "a_is_signed": false,
    "i_partition": 0,
    "i_is_replica": false,
    "i_vstamp": 1,
    "r_folder_path": [
        "/DeeShan/Students/Instance3"
    ],
    "i_ancestor_id": [
        "810003e98000210b",
        "0b0003e980005965",
        "0c0003e980005a61"
    ],
    "template_id": "080003e980005d63",
    "firstname": "Jane",
    "lastname": "Doe",
    "course": "MBBS",
    "r_object_id": "810003e98000210b",
    "bws_type_def_description": "This is a description for Type Def of
Student",

```

```
        "bws_type_def_label_name": "Student"
    },
    "links": [
        {
            "rel": "self",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/business-workspaces/810003e98000210b"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/parent-links",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/parent-links"
        },
        {
            "rel": "parent",
            "title": "0b0003e980005965",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/folders/0b0003e980005965"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/cabinet",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/cabinets/0c0003e980005a61"
        },
        {
            "rel": "contents",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/contents"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/primary-content",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/contents/content"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/checkout",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/lock"
        },
        {
            "rel": "version-history",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/versions"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/current-version",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/versions/current"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/virtual-document-nodes",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/objects/810003e98000210b/vd-nodes"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/snapshots",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/snapshots?filter=r_assembled_from_id%20%3D'810003e98000210b'"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/audit-trails",
            "href": "http://localhost:8183/documentum-rest-web/repositories/d2repo/audit-trails"
        }
    ]
}
```

```

d2repo/audit-trails?filter=audited_obj_id='810003e98000210b'"
},
{
    "rel": "edit",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b"
},
{
    "rel": "http://identifiers.emc.com/linkrel/delete",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b"
},
{
    "rel": "http://identifiers.emc.com/linkrel/folders",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/folders"
},
{
    "rel": "http://identifiers.emc.com/linkrel/documents",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/documents"
},
{
    "rel": "http://identifiers.emc.com/linkrel/objects",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/objects"
},
{
    "rel": "http://identifiers.emc.com/linkrel/child-links",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/folders/810003e98000210b/child-links"
},
{
    "rel": "http://identifiers.emc.com/linkrel/relations",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/relations?related-object-id=810003e98000210b&related-object-role=any"
},
{
    "rel": "http://identifiers.emc.com/linkrel/object-aspects",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/aspects"
},
{
    "rel": "http://identifiers.emc.com/linkrel/permission-set",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/permission-set"
},
{
    "rel": "http://identifiers.emc.com/linkrel/permissions",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/permissions"
},
{
    "rel": "http://identifiers.emc.com/linkrel/comments",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/comments"
},
{
    "rel": "http://identifiers.emc.com/linkrel/attachable-lifecycles",
    "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/lifecycles?object-id=810003e98000210b"
},

```

```
        {
            "rel": "http://identifiers.emc.com/linkrel/object-lifecycle",
            "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/objects/810003e98000210b/lifecycle"
        },
        {
            "rel": "http://identifiers.emc.com/linkrel/bws-roles",
            "href": "http://localhost:8183/documentum-rest-web/repositories/
d2repo/business-workspaces/810003e98000210b/roles"
        }
    ]
}
```



### 2.95.3 Get List of Workspace Roles by Workspace Instance ID

Retrieves the list of Workspace Roles which is created at runtime in the `dm_group` type.

#### 2.95.3.1 URI

- Item: Base URI
- Value: `{repositoryUri}/repositories/{repositoryName}/business-
workspaces/{instanceid}/roles`

#### 2.95.3.2 Navigation

Workspaces

#### 2.95.3.3 Link relations

`http://identifiers.emc.com/linkrel/bws-roles`

#### 2.95.3.4 HTTP Request method

GET

#### 2.95.3.5 Query parameters

Supports the following common query parameters:

- `inline`
- `links`
- `view`
- `page`
- `filter`
- `items-per-page`

For more information, see [Appendix B, REST Common Definition - URI Request Query Parameters](#) on page 1047.

#### 2.95.3.6 Request media types

- application/vnd.emc.documentum+json
- application/json

#### 2.95.3.7 Request headers

- Accept
- Authorization

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.95.3.8 Request body

N/A

#### 2.95.3.9 Response headers

Content-Type

For more information, see [Appendix D, REST Common Definition - HTTP Headers](#) on page 1055.

#### 2.95.3.10 Response media types

- application/vnd.emc.documentum+json
- application/json

#### 2.95.3.11 Response status

- 200 - Retrieved successfully
- 400, 401, 403, 404, 406, 409, 415, 500, and so on

For more information, see [Appendix C, REST Common Definition - HTTP Status Codes](#) on page 1053.

### 2.95.3.12 Response body

JSON representation.

#### Example 2-488: JSON response

```
{  
    "id": "http://localhost:8080/dctm-rest/repositories/d2repo/business-  
workspaces/810003e98000010a/roles",  
    "author": [  
        {  
            "name": "Open Text Documentum"  
        }  
    ],  
    "updated": "2025-07-24T09:50:17.421+00:00",  
    "page": 1,  
    "items-per-page": 100,  
    "links": [  
        {  
            "rel": "self",  
            "href": "http://localhost:8080/dctm-rest/repositories/d2repo/  
business-workspaces/810003e98000010a/roles?link=true&inline=true"  
        }  
    ],  
    "entries": [  
        {  
            "id": "120003e98000110c",  
            "links": [  
                {  
                    "rel": "edit",  
                    "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/admin_bws_810003e98000010a"  
                }  
            ],  
            "content": {  
                "name": "bws-role",  
                "definition": "http://localhost:8080/dctm-rest/repositories/  
d2repo/types/dm_group",  
                "properties": {  
                    "r_object_id": "120003e98000110c",  
                    "group_name": "admin_bws_810003e98000010a",  
                    "owner_name": "dmadmin",  
                    "is_private": false,  
                    "description": "This is an admin role",  
                    "globally_managed": false,  
                    "alias_set_id": "0000000000000000",  
                    "group_class": "bws",  
                    "group_admin": "Admin",  
                    "is_dynamic": false,  
                    "is_dynamic_default": false,  
                    "group_global_unique_id":  
                        "d2repo:admin_bws_810003e98000010a",  
                    "group_native_room_id": "810003e98000010a",  
                    "group_directory_id": "0000000000000000",  
                    "is_protected": false,  
                    "is_module_only": false,  
                    "r_modify_date": "2025-07-23T09:56:02.000+00:00",  
                    "i_vstamp": 0,  
                    "group_display_name": "Administrator"  
                },  
                "links": [  
                ]  
            }  
        }  
    ]  
}
```

```

        "rel": "self",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admin_bws_810003e98000010a"
    },
    {
        "rel": "edit",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admin_bws_810003e98000010a"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/delete",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admin_bws_810003e98000010a"
    },
    {
        "rel": "parent",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups?group-name=admin_bws_810003e98000010a"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/groups",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admin_bws_810003e98000010a/groups"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/users",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admin_bws_810003e98000010a/users"
    },
    {
        "rel": "http://identifiers.emc.com/linkrel/members",
        "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/admin_bws_810003e98000010a/members"
    }
]
},
{
    "id": "120003e98000110d",
    "links": [
        {
            "rel": "edit",
            "href": "http://localhost:8080/dctm-rest/repositories/
d2repo/groups/applicant_bws_810003e98000010a"
        }
    ],
    "content": {
        "name": "bws-role",
        "definition": "http://localhost:8080/dctm-rest/repositories/
d2repo/types/dm_group",
        "properties": {
            "r_object_id": "120003e98000110d",
            "group_name": "applicant_bws_810003e98000010a",
            "owner_name": "dmadmin",
            "is_private": false,
            "description": "This is an applicant role",
            "globally_managed": false,
            "alias_set_id": "0000000000000000",
            "group_class": "bws",
            "group_admin": "Admin",
            "is_dynamic": false,
            "is_dynamic_default": false,
            "is_group": true
        }
    }
}

```

```
        "group_global_unique_id":  
        "d2repo:applicant_bws_810003e98000010a",  
        "group_native_room_id": "810003e98000010a",  
        "group_directory_id": "0000000000000000",  
        "is_protected": false,  
        "is_module_only": false,  
        "r_modify_date": "2025-07-23T09:56:02.000+00:00",  
        "i_vstamp": 0,  
        "group_display_name": "Applicant"  
    },  
    "links": [  
        {  
            "rel": "self",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/applicant_bws_810003e98000010a"  
        },  
        {  
            "rel": "edit",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/applicant_bws_810003e98000010a"  
        },  
        {  
            "rel": "http://identifiers.emc.com/linkrel/delete",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/applicant_bws_810003e98000010a"  
        },  
        {  
            "rel": "parent",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups?group-name=applicant_bws_810003e98000010a"  
        },  
        {  
            "rel": "http://identifiers.emc.com/linkrel/groups",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/applicant_bws_810003e98000010a/groups"  
        },  
        {  
            "rel": "http://identifiers.emc.com/linkrel/users",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/applicant_bws_810003e98000010a/users"  
        },  
        {  
            "rel": "http://identifiers.emc.com/linkrel/members",  
            "href": "http://localhost:8080/dctm-rest/repositories/  
d2repo/groups/applicant_bws_810003e98000010a/members"  
        }  
    ]  
}  
]
```



# Appendix A. Link Relations

Foundation REST API uses the following link relations:

**Table A-1: Public link relations**

Link relation	Description	Specification
about	Returns product information	RFC6903
canonical	Designates an Internationalized Resource Identifier (IRI) as preferred over resources with duplicative content.	RFC6596
child	Points to a hierarchical child, or subobject, of the current object.	
contents	Points to the contents metadata feed for a content object.	HTML 4.01 specification
edit	Points to a resource that can be used to edit the link's context.	RFC5023
enclosure	Identifies a related resource that is potentially large and might require special handling.	RFC5023
first	An IRI that refers to the furthest preceding resource in a series of resources.	RFC5005
icon	Points to the icon for an entry, a page or a site.	WHATWG website
last	An IRI that refers to the furthest following resource in a series of resources.	RFC5005
next	Indicates that the link's context is a part of a series, and that the next in the series is the link target.	RFC5005
parent	Refers to one of the following: <ul style="list-style-type: none"><li>• parent type of a type</li><li>• parent folder of a SysObject</li><li>• parent document of a document</li><li>• parent object of a relation</li><li>• parent group of a group, a role, or a user.</li></ul>	
predecessor-version	Points to a resource containing the predecessor version in the version history.	RFC5829

Link relation	Description	Specification
previous	Points to the previous resource in an ordered series of resources.	RFC5005
related	Points to the feed for a SysObject related to specified relation type(s).	RFC4287
self	Conveys an identifier for the link's context.	RFC4287
version-history	Points to a resource containing the version history for the context.	RFC5829

**Table A-2: Link relations introduced in Foundation REST API**

Link relation	Description
acl	Points to the ACL resource for a specific permission set object
acls	Points to the ACLs feed under a repository
as-search-template	Points to the save as search template behavior for a saved search
associations	Points to the associated Sysobjects feed for a specific ACL object
assist-values	Points to the type assist values resource to obtain the value assistance of a type
aspect-types	Points to the aspect types resource to show all aspect types of a repository
aspect-type	Points to the single aspect type resource
batches	Points to the batches resource under a repository.
batch-capabilities	Points to the batch capabilities resource under a repository.
cabinets	Points to the cabinets feed under a repository.
cancel-checkout	Points to the cancel-checkout behavior for a versionable object.
checkin-branch	Points to the checkin as branch version behavior for an object that is checked out.
checkin-next-major	Points to the checkin as next major version behavior for an object that is checked out.
checkin-next-minor	Points to the checkin as next minor version behavior for an object that is checked out.
checkout	Points to the checkout behavior for a versionable object .
child-links	Points to the child links feed resource of a folder.
content-media	Indicates that the content can be downloaded.
comments	Points to top level comments for a SysObject.
current-user	Points to the current login user resource under a repository.
current-user-preferences	Points to the current user preferences resource to manipulate the user preference

<b>Link relation</b>	<b>Description</b>
current-version	Points to the current version resource for a versionable object.
default-folder	Points to the default folder resource for a user.
delete	Points to the delete behavior of an object.
dematerialize	link to dematerialize a lightweight object
documents	Points to the documents feed under a repository.
folders	Points to the folders feed under a repository.
format	Points to the format resource for a content resource.
formats	Points to the formats feed under a repository.
groups	Points to the groups feed under a repository or direct member groups under a group.
lightweight-types	Points to the types resource which contains the collection of children lightweight types of a shareable type
lightweight-objects	Points to the collection of lightweight objects shares one shareable object
logoff	Points to the log out action for single sign on user.
materialize	link to materialize a lightweight object
objects	lists folder contents.
object-aspects	Points to the object aspects resource to get attached aspects of an object, or attache an aspect to the object
parent-links	Points to the parent links resource of a non-cabinet SysObject.
parent-shareable-type	Points to the parent shareable type of a lightweight type
permissions	Points to the permissions resource for a Sysobject
permission-set	Points to the permission set resource for a Sysobject or a user
primary-content	Points to the primary content resource for a content SysObject type.
relate	Indicates that the object can be related to other objects.
relation-type	Points to the relation type resource for a relation instance.
relation-types	Points to the relation types feed under a repository.
relations	Points to the relations feed under a repository, relations feed for a specific relation type, or relations feed related to a specified object.
replies	Points to replied comments for a parent comment.
repositories	Points to the repositories feed from the docbrokers.
saved-searches	Points to the saved searches resource under a repository
search-templates	Points to the search templates resource under a repository

Link relation	Description
search-execution	Points to the execution behavior for a saved search or search template
saved-search-results	Points to the results resource of a saved search
shared-parent	Points the parent shareable object of a lightweight object
types	Points to the data dictionary types feed under a repository.
users	Points to the users feed under a repository, or direct member users under a group.
virtual-document-nodes	Points to the virtual document nodes resource to fetch all nodes of an virtual document
These link relations are defined by OpenText Documentum CM. The fully qualified OpenText Documentum CM link relation path is prefixed with the following string: <a href="http://identifiers.emc.com/linkrel/">http://identifiers.emc.com/linkrel/</a> .	

## Appendix B. REST Common Definition - URI Request Query Parameters

Table B-1: URI request query parameters

Query parameter	Description	Data type	Value range	Default value
dql	a DQL query	DQL expression	N/A	N/A
filter	A filter expression in a subset of XPath  For more information, see <i>OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)</i> .	N/A	N/A	no filtering
include-total	Calculates the total number of feed items instead of returning them all in one page	string	<i>true, false</i>  When <i>true</i> the total number of feed items are calculated by the server and returned  When <i>false</i> no calculation of the total count is done	<i>false</i>

Query parameter	Description	Data type	Value range	Default value
inline	Specifies whether or not to show content (the object instance) in the atom entry for a collection	string	<p><i>true, false</i></p> <p>When set to <i>true</i> it returns the object instance embedded into the entry's content element</p> <p>When <i>false</i> it does not return an object instance</p>	<p><i>false</i></p> <p> <b>Note:</b> Although the default value is <i>false</i>, we recommend that you try setting this parameter to <i>true</i> in your development environment to get the entire content of resources and explore the complete data structure.</p> <p>In your production environment, you can set this parameter to <i>false</i> for better performance.</p>
items-per-page	<p>The number of entries per page</p> <p>When the result feed is paged, an element <code>&lt;dm:items-per-page&gt;xx&lt;/dm:items-per-page&gt;</code> is displayed under the feed root</p>	integer	<code>&gt;=1</code>	<code>100</code>
links	Specifies whether or not the link relations are returned for this object representation	boolean	<p><i>true, false</i></p> <p>A value of <i>false</i> means that no links that are too expensive to calculate are returned. Only easy to navigate links are returned</p>	<i>true</i>

Query parameter	Description	Data type	Value range	Default value
page	The number of the page to be served  When the result feed is paged, an element <code>&lt;dm:page&gt;xx&lt;/dm:page&gt;</code> is displayed under the feed root	integer	<code>&gt;=1</code>  When set to <code>true</code> , an element <code>&lt;dm:total&gt;xx&lt;/dm:total&gt;</code> is displayed under the feed root	<code>1</code>  For example when <code>item_per_page=200</code> <code>&amp;page=2</code>  Returns items number 201 to 400
q	Specifies full text search criterion when sending a GET request to the <b>Search</b> resource or certain collection resources  For more information, see <i>OpenText Documentum Content Management - Foundation REST API Development Guide</i> ( <i>EDCPKRST250400-PGD</i> ).	string	String that follows the Simple Search Language syntax	N/A
recursive	When get children of an object, whether include all indirect children recursively, e.g. children's children	boolean	<code>true, false</code>	<code>false</code>

Query parameter	Description	Data type	Value range	Default value
sort	<p>Sorting collection result entries. This parameter can also be used to perform case insensitive sorting.</p> <ul style="list-style-type: none"> <li>• <i>sort=object_name:</i> Sort by attribute &lt;object_name&gt; in ascending order, which is the default sort order.</li> <li>• <i>sort=object_name asc no-case:</i> Sort by attribute &lt;object_name&gt; in ascending and case insensitive order.</li> <li>• <i>sort=r_modify_date desc, object-id asc no-case:</i> Sort primarily by attribute &lt;r_modify_date&gt; in descending order and secondarily by attribute &lt;object_name&gt; in ascending case insensitive order.</li> </ul>	string	<p>Defining a sort consists of using multiple sort specifications, separated by comma (',') character.</p> <p>Each sort specification consists of an attribute to be sorted and its sort order, separated by a space (' ') character.</p> <p>Sort results can be returned in either &lt;DESC&gt; or &lt;ASC&gt; sort order. Using a sort order is optional and when a sort order is not specified, the default sort order, which is &lt;ASC&gt;, gets used.</p> <p>You can use the optional &lt;no-case&gt; parameter to perform a case insensitive sort.</p> <p>You also have the option to sort with non-repeating attributes. For example: <code>sort=r_modify_date desc,object_id asc,title</code></p> <p><b>! Important</b> When an attribute with an invalid name is specified, an error is thrown.</p>	<p>Varies by resource and each resource has its own default order.</p> <p><b>Notes</b></p> <ul style="list-style-type: none"> <li>• Whether the default repository supports case sensitive sort or not is depending on the database installation. In REST, you can use the sort parameter to make the sort mode case insensitive, but you cannot use this parameter to make the sort mode case sensitive.</li> <li>• Using the no-case parameter has side effects on non-string attributes. For example, <code>sort=r_creation_date desc no-case</code> sorts the date time attribute in string</li> </ul>

Query parameter	Description	Data type	Value range	Default value
				value order, which may not be what is expected.
view	<p>Properties to return</p> <p>For more information, see <i>OpenText Document Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)</i>.</p>	<p>string</p> <p>The view-name and columns* parameters must be mutually exclusive.</p> <p>Names of predefined views start with a colon ":" character.</p> <p>The following view-name values are defined:</p> <ul style="list-style-type: none"> <li>• all</li> <li>• default</li> </ul> <p>When no view-name is specified, the names of properties or predefined views are returned as a comma separated list.</p>	<p>For example, ?view=[view name]?(.column)*</p>	default



## Notes

- Please see the individual resource sections to see how each of the above parameters are used, as this may vary by resource.
- When the `view` parameter is used on a single object resource that supports this parameter (such as the Cabinet resource, Document resource, and so on), the `view` attribute names must match the definition of the object data type. A `Bad request (400)` error is returned if any unknown attributes for the object data type are specified in the `view` parameter.

When the `view` parameter is used on a collection based resource that supports this parameter (such as the Cabinets resource, Folder Child Documents resource, etc.), a comma separated list of `view` attributes can contain both base type attribute names and arbitrary extended attribute names.

For example, let's assume that a folder X has a number of documents in it with the object type `<dm_document>` and document sub types `<doc_ext1>`, `<doc_ext2>`, and so on. When a REST client retrieves the folder children, it can specify the `view` attributes from both `<dm_document>` and its sub types.

Here is a code sample that shows you the request:

```
// GET a folder child documents feed.  
// In this sample, attributes 'r_object_id' and 'object_name' come from dm_document.  
// 'doc_ext1_attr' comes from doc_ext1, and 'doc_ext2_attr' comes from dmc_ext2.  
GET /documents?inline=true&  
view=r_object_id,object_name,doc_ext1_attr,doc_ext2_attr HTTP/1.1
```

Please also note the following `view` usage on a collection based resource:

- Aspect type attributes can also be put into the `view` when the aspect type is attachable to the collection base type, such as `.<dm_document>` is in the above sample.
- Collection items return the specified attributes only when the attributes exist in those objects.
- Unknown attributes for collection items are ignored.

## Appendix C. REST Common Definition - HTTP Status Codes

Table C-1: HTTP Status Codes

HTTP Status	Description
200 OK	The Request has succeeded. The information returned with the response is dependent on the method used in the Request, for example: GET an entity corresponding to the requested resource is sent in the response; PUT an entity describing or containing the modified resource
201 Created	The Request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field. The entity format is specified by the media type given in the Content-Type header field
204 No Content	The Server fulfilled the Request, but does not need to return a response message body
304 Not Modified	The Request performs a GET on the resource, but the resource has not been modified since last GET
3xx Redirections	Future use
400 Bad Request	The Request could not be understood by the server due to malformed syntax, missing or invalid information (such as validation error on an input field, a missing required value, and so on). The client SHOULD NOT repeat the Request without modifications
401 Unauthorized	The authentication credentials included with this request are missing or invalid. The Response MUST include a WWW-Authenticate header field containing a challenge applicable to the requested resource
403 Forbidden	The Server recognized your credentials, but you do not possess authorization to perform this request, and the Request SHOULD NOT be repeated
404 Not Found	The Request specified a URI of a resource that does not exist
405 Method Not Allowed	The HTTP verb specified in the Request (DELETE, GET, HEAD, POST, PUT) is not supported for this request URI
406 Not Acceptable	The resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept header of the Request

HTTP Status	Description
409 Conflict	A creation or update request could not be completed, because it would cause a conflict in the current state of the resources supported by the server (for example, an attempt to create a new resource with a unique identifier already assigned to some existing resource)
415 Unsupported Media Type	The Server is refusing to service the Request because the entity of the Request is in a format not supported by the requested resource for the requested method
500 Internal Server Error	The Server encountered an unexpected condition which prevented it from fulfilling the Request

## Appendix D. REST Common Definition - HTTP Headers

**Table D-1: HTTP headers**

HTTP header	Description	In request or response?	Value range	
Accept-Language	Indicates the preferred language(s) for error messages	Request	A language range complying with RFC4647.	
Authorization	The authorization header for authentication	Request	BASE64 encoded HTTP basic authentication header with encoded credential.  For example:  Authorization: Basic QWxhZGRpbjpvcGVu IHNlc2FtZQ==  Authorization: Negotiate YIIZG1hZG1p bjpwYXNzd29yZ...  	
Accept	Acceptable media type for the response	Request	See note at the end of this table.	
Content-Type	Media type of the request or response body	Request Response	See note at the end of this table.	
Content-Disposition	Disposition header of the response body  Content-Disposition header parameters include: fileName and fileName*  If the charset of the content filename that you want to download belongs to ISO_8859_1, then the fileName header parameter is used. Otherwise, the fileName* header parameter is used.	Response	Complies with RFC5987 and RFC6266.	
Content-Length	The length of the request body in octets (8-bit bytes)	Request Response	Non-negative number	
Location	URI of the newly created resource	Response	URI	

HTTP header	Description	In request or response?	Value range
ETag	ETag value generated by REST server	Response	String value
If-None-Match	Checks whether the resource has changed	Request	ETag value in previous server's response
Last-Modified	Last modified time of the resource	Response	HTTP-Date
If-Modified-Since	Checks whether the resource has changed since last modified time	Request	Last-modified value in the previous server's response
Set-Cookie	Set an HTTP cookie	Response	Used by client token
WWW-Authenticate	Indicates the authentication scheme that should be used to access the requested entity	Response	Used by HTTP basic authentication



### Notes

- The Accept request-header field can be used to specify certain media types which are acceptable for the response. Accept headers can be used to indicate that the request is specifically limited to a small set of desired types, as in the case of a request for an in-line image.

The asterisk "\*" character is used to group media types into ranges, with "\*/\*" indicating all media types and "type/\*" indicating all subtypes of that type. The media-range MAY include media type parameters that are applicable to that range.

Each media-range can be followed by one or more accept-params, beginning with the "q" parameter for indicating a relative quality factor. The first "q" parameter (if any) separates the media-range parameter(s) from the accept-params. Quality factors allow the user or user agent to indicate the relative degree of preference for that media-range, using the qvalue scale from 0 to 1. The default value of q is 1.



### Example D-1: Example

```
Accept: audio/*; q=0.2, audio/basic
```

The above statement is interpreted as "I prefer audio/basic, but send me any audio type if it is the best available after an 80% reduction in quality."



- The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient, or in the case of the HEAD method, the media type that would have been sent had the Request been a GET.

 **Example D-2: Example**

```
Content-Type = "Content-Type" ":" media-type
```

Specify a value for media-type above. Here is an example of a field:

```
Content-Type: text/html; charset=ISO-8859-4
```





## Appendix E. AQL

The Abstract Query Language (AQL) is an implementation-independent search description language. The language hides the actual query language used for OpenText Documentum CM components (DQL, xQuery) or external sources. In REST services, the Search resource utilizes AQL to represent its query document.

### ➡ Example E-1: Query document in AQL - XML

```
<?xml version='1.0' encoding='UTF-8'?>
<search all-versions="true" max-results-for-facets="10"
xmlns="http://identifiers.emc.com/vocab/documentum"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <types>
        <type>dm_document</type>
    </types>
    <columns>
        <column>r_object_type</column>
        <column>object_name</column>
    </columns>
    <sorts>
        <sort ascii="false" ascending="true" lang="fr">r_content_size</sort>
        <sort ascii="true" ascending="false" lang="zh">object_name</sort>
    </sorts>
    <collections>
        <collection>collection1</collection>
        <collection>collection2</collection>
    </collections>
    <locations>
        <id-location descendant="true">
            <id>0c00000580001914</id>
        </id-location>
        <path-location descendant="false">
            <path>/dadmin</path>
        </path-location>
    </locations>
    <facet-definitions>
        <facet-definition id="id1" group-by="alpharange" max-values="8">
            <attributes>
                <attribute>r_object_type</attribute>
            </attributes>
            <sort>FREQUENCY</sort>
            <properties>
                <property name="range">a:e, f:z</property>
            </properties>
        </facet-definition>
        <facet-definition id="id2" group-by="relativeDate" max-values="9">
            <attributes>
                <attribute>r_modify_date</attribute>
            </attributes>
            <sort>VALUE_ASCENDING</sort>
        </facet-definition>
    </facet-definitions>
    <expression-set>
        <expressions>
            <property name="object_name" operator="CONTAINS" exact-
match="true"
                    repeating="true" fuzzy="true">rest
        </expressions>
    </expression-set>
</search>
```

```

        </property>
        <fulltext fuzzy="true">rest</fulltext>
        <fulltext fuzzy="false">test</fulltext>
        <property-range name="r_content_size" operator="BETWEEN"
repeating="false">
            <from>0</from>
            <to>10000</to>
        </property-range>
        <property-list name="r_object_type" operator="IN"
repeating="false">
            <values>
                <value>dm_document</value>
                <value>dm_folder</value>
            </values>
        </property-list>
        <expression-set>
            <expressions>
                <fulltext fuzzy="false">test</fulltext>
            </expressions>
        </expression-set>
        <relative-date name="r_modify_date" time-unit="MONTH"
operator="GREATER THAN" >-2</relative-date>
    </expressions>
</expression-set>
</search>

```



#### ➡ Example E-2: Query document in AQL - JSON

```
{
    "all-versions": true,
    "include-hidden-objects": false,
    "max-results-for-facets": 10,
    "types": [
        "dm_document"
    ],
    "columns": [
        "r_object_type",
        "object_name"
    ],
    "sorts": [
        {
            "property": "r_content_size",
            "ascending": true,
            "lang": "fr",
            "ascii": false
        },
        {
            "property": "object_name",
            "ascending": false,
            "ascii": true
        }
    ],
    "collections": [
        "collection1",
        "collection2"
    ],
    "locations": [
        {
            "location-type": "id-location",
            "id": "0c00000580001914",
            "repository": "REPO",

```

```
        "descendant": true
    },
    {
        "location-type": "path-location",
        "path": "/dmadmin",
        "repository": "REPO",
        "descendant": false
    }
],
"facet-definitions": [
{
    "id": "id1",
    "attributes": [
        "r_object_type"
    ],
    "group-by": "alpharange",
    "sort": "FREQUENCY",
    "properties": {
        "skipEmptyValues": "true",
        "range": "a:e, f:z"
    },
    "max-values": 8
},
{
    "id": "id2",
    "attributes": [
        "r_modify_date"
    ],
    "group-by": "relativeDate",
    "sort": "VALUE_ASCENDING",
    "properties": {
        "returnUTC": "true",
        "skipEmptyValues": "true"
    },
    "max-values": 9
}
],
"repositories": [
    "REPO"
],
"expression-set": {
    "expression-type": "expression-set",
    "operator": "AND",
    "expressions": [
        {
            "expression-type": "property",
            "name": "object_name",
            "operator": "CONTAINS",
            "value": "rest",
            "exact-match": true,
            "repeating": true,
            "case-sensitive": true,
            "fuzzy": true
        },
        {
            "expression-type": "fulltext",
            "value": "rest",
            "fuzzy": true
        },
        {
            "expression-type": "fulltext",
            "value": "test",
            "fuzzy": false
        },
        {
            "expression-type": "property-range",
            "name": "r_content_size",
            "operator": "BETWEEN",
            "from": "0",
            "to": "10000"
        }
    ]
}
```

```

        "repeating": false
    },
    {
        "expression-type": "property-list",
        "name": "r_object_type",
        "operator": "IN",
        "values": [
            "dm_document",
            "dm_folder"
        ],
        "repeating": false
    },
    {
        "expression-type": "expression-set",
        "operator": "AND",
        "expressions": [
            {
                "expression-type": "fulltext",
                "value": "test",
                "fuzzy": false
            }
        ]
    },
    {
        "expression-type": "relative-date",
        "name": "r_modify_date",
        "value": -2,
        "time-unit": "MONTH",
        "operator": "GREATER_THAN",
        "repeating": false
    }
]
}

```



The following table contains AQL elements.

Element	Description
all-versions	Indicates whether or not all versions of the objects are expected to be included in the query results.
include-hidden-objects	Indicates whether or not the hidden objects should be included in the query results. Its default value is false.
types	Defines the type of the objects to search. You cannot define multiple types. The URL parameter <code>object-type</code> cannot be used in a search operation with AQL.
columns	Defines the properties of the object to display in the result set. This element works when the URL parameter <code>inline</code> is set to true. The URL parameter <code>view</code> cannot be used in a search operation with AQL.

Element	Description				
collections	<p>Defines the collections to search.</p> <p>The URL parameter <code>collections</code> cannot be used in a search operation with AQL.</p>				
locations	<p>Defines the folders to search.</p> <p>You can specify locations in the following two ways:</p> <p>The URL parameter <code>locations</code> cannot be used in a search operation with AQL.</p>				
Location Type	Option	Example			
id location	<p>id: The ID of the folder to search</p> <p>descendent: Whether to search the sub-folders or not</p> <p>Default value: false</p>	<pre>&lt;id-location descendent="true"&gt; &lt;id&gt;0c00000580001914&lt;/id&gt; &lt;/id-location&gt;</pre>			
path location	<p>path: The path of the folder to search</p> <p>descendent: Whether to search the sub-folders or not</p> <p>Default value: false</p>	<pre>&lt;path-location descendent="false"&gt; &lt;path&gt;/dmadmin&lt;/path&gt; &lt;/path-location&gt;</pre>			
The URL parameter <code>locations</code> cannot be used in a search operation with AQL.					
sorts	<p>Defines how the system sorts results.</p> <p>REST services uses two strategies, ASCII sort and locale sort. The following table describes the details:</p>				
Option	Description				
property	Property to sort against				
ascending	<p>Specifies whether or not to sort the results in ascending order</p> <p>Default value: true</p>				
ascii	<p>Specifies whether or not to sort items using the ASCII table.</p> <p>Default value: true</p> <p>When this value is set to false, you must specify a locale using <code>lang</code>.</p>				

Element	Description	
	lang	Specifies the locale that the system uses to sort the results. The supported locale depends on the xPlore configuration.
	You can specify multiple sort items. In this case, the system sorts results by the first sort item first, the by the second one, and so on so forth. For example, with the sorts element defined as follows, the system sorts results by <code>&lt;r_content_size&gt;</code> first, and then by <code>&lt;object_name&gt;</code> for objects with the same <code>&lt;r_content_size&gt;</code> .	
expression-set	<p>Defines the search criteria by using a set of expressions. Expression-sets can be nested and associated with each other using logical AND (default) or logical OR. The following five expressions are supported:</p> <ul style="list-style-type: none"> <li>• fulltext expression</li> <li>• property expression</li> <li>• property list expression</li> <li>• property range expression</li> <li>• relative date expression</li> </ul>	
facet-definitions	<p>List of facet definitions. See <a href="#">Facet in AQL</a>.</p> <p>The URL parameters <code>facet</code> and <code>facet-value-constraints</code> cannot be used in a search operation with AQL.</p>	
max-results-for-facets	Sets the maximum number of results to return.	

## E.1 Expressions in AQL

The `expression-set` element defines the search criteria by using a set of expressions. Expressions can be nested and associated with each other using logical AND (default) or logical OR.

### Example E-3: XML

```

<expression-set>
  <expressions>
    <property name="object_name" operator="CONTAINS" exact-match="true"
      repeating="true" fuzzy="true">rest
    </property>
    <fulltext fuzzy="true">rest</fulltext>
    <fulltext fuzzy="false">test</fulltext>
    <property-range name="r_content_size" operator="BETWEEN" repeating="false">
      <from>0</from>
      <to>10000</to>
    </property-range>
    <property-list name="r_object_type" operator="IN" repeating="false">
      <values>
        <value>dm_document</value>
        <value>dm_folder</value>
      </values>
    </property-list>
    <expression-set>
      <expressions>
        <fulltext case-sensitive="false" fuzzy="false">test</fulltext>
      </expressions>
    </expression-set>
  </expressions>
</expression-set>

```

```

        </expression-set>
        <relative-date name="r_modify_date" time-unit="MONTH"
operator="GREATER_THAN" >-2</relative-date>
    </expressions>
</expression-set>

```



### Example E-4: JSON

```

"expression-set": {
    "expression-type": "expression-set",
    "operator": "AND",
    "expressions": [
        {
            "expression-type": "property",
            "name": "object_name",
            "operator": "CONTAINS",
            "value": "rest",
            "exact-match": true,
            "repeating": true,
            "case-sensitive": true,
            "fuzzy": true
        },
        {
            "expression-type": "fulltext",
            "value": "rest",
            "fuzzy": true
        },
        {
            "expression-type": "fulltext",
            "value": "test",
            "fuzzy": false
        },
        {
            "expression-type": "property-range",
            "name": "r_content_size",
            "operator": "BETWEEN",
            "from": "0",
            "to": "10000",
            "repeating": false
        },
        {
            "expression-type": "property-list",
            "name": "r_object_type",
            "operator": "IN",
            "values": [
                "dm_document",
                "dm_folder"
            ],
            "repeating": false
        },
        {
            "expression-type": "expression-set",
            "operator": "AND",
            "expressions": [
                {
                    "expression-type": "fulltext",
                    "value": "test",
                    "fuzzy": false
                }
            ]
        },
        {
            "expression-type": "relative-date",
            "name": "r_modify_date",
            "value": -2,
            "time-unit": "MONTH",

```

```
        "operator": "GREATER_THAN",
        "repeating": false
    }
}
```



## E.1.1 Fulltext expression

A fulltext expression is similar to the URL parameter q in that it follows the Simple Search Language syntax. Moreover, fulltext expressions allow you to perform a fuzzy search by using the fuzzy parameter, which takes a boolean value and defaults to *true*. A fuzzy search is a process that returns values based on approximate string value matching. In other words, the criteria used to perform a fuzzy search does not have to match exactly to be relevant enough to return search results.

In the following fuzzy search examples, the first row, where fuzzy search is set to *false*, will match RESTful services, but not RESTFULL or RESTfol. The search criteria must have an exact match to return a result.

The second row, where fuzzy is set to *true*, matches all of the above terms.

### Example E-5: Expressions using the fuzzy search parameter

```
<!-- Fuzzy search set to false -->
<fulltext fuzzy="false">RESTful</fulltext>

<!-- Fuzzy search set to true -->
<fulltext fuzzy="true">RESTful</fulltext>
```



## E.1.2 Property expression

A property expression defines property value constraints by using a variety of logical operators.

**Table E-1: Property expression**

Option	Description
name	Property name
value	Property value
operator	Supports the following logical operators: EQUAL, NOT_EQUAL, GREATER_THAN, LESS_THAN, GREATER_EQUAL, LESS_EQUAL, BEGINS_WITH, CONTAINS, DOES_NOT_CONTAIN, ENDS_WITH, IS_NULL, IS_NOT_NULL

Option	Description
fuzzy	Whether or not to enable fuzzy search for the following operators: SEARCH_OP_DOES_NOT_CONTAIN, SEARCH_OP_CONTAINS, SEARCH_OP_EQUAL  Default value: <i>false</i>
exact-match	Whether or not to evaluate an EQUAL or NOT_EQUALS constraint.  When the constraint is evaluated, the system performs no stemming, stop words removal, thesaurus search, or wildcard matching.  Default value: <i>false</i>
repeating	Determines whether or not Foundation Java API should generate DQL instead of xQuery when the type of a repeating property is neither DF_ID nor DF_STRING.  Default value: <i>false</i>
case-sensitive	Indicates whether the search operation is case sensitive. Default value: <i>false</i>

The following property expression returns *true* when the object\_name of an object is rest.

➡ **Example E-6: Property expression in XML**

```
<property name="object_name" operator="EQUAL" exact-match="true"
          repeating="true" case-sensitive="true" fuzzy="false">rest
</property>
```



➡ **Example E-7: Property expression in JSON**

```
{
    "expression-type": "property",
    "name": "object_name",
    "operator": "EQUAL",
    "value": "rest",
    "exact-match": true,
    "repeating": true,
    "case-sensitive": true,
    "fuzzy": false
}
```



### E.1.3 Property list expression

A property expression defines property value constraints by using a list of discrete values. The following table describes the details:

**Table E-2: Property list expression**

Option	Description
name	Property name
operator	Supports the following logical operators: <ul style="list-style-type: none"> <li>• IN: returns true if the value of the property is in the value list.</li> <li>• NOT IN: returns true if the value of the property is not in the value list.</li> </ul>
values	List of discrete property values
repeating	Whether the property is repeating

The following expression returns true if the `r_object_type` is either `dm_document` or `dm_folder`.

➡ **Example E-8: Property list expression in XML**

```
<property-list name="r_object_type" operator="IN" repeating="false">
  <values>
    <value>dm_document</value>
    <value>dm_folder</value>
  </values>
</property-list>
```



➡ **Example E-9: Property list expression in JSON**

```
{
  "expression-type": "property-list",
  "name": "r_object_type",
  "operator": "IN",
  "values": [
    "dm_document",
    "dm_folder"
  ],
  "repeating": false
}
```



## E.1.4 Property range expression

A property expression defines property value constraints by using a value range. The following table describes the details:

**Table E-3: Property range expression**

Option	Description
name	Property name
operator	Supports the logical BETWEEN operator. Returns <i>true</i> when the value of the property is in the range specified by from and to attributes.
from	Starting value
to	Ending value
repeating	Whether the property is repeating

The following expression returns true if the r\_content\_size is smaller than or equal to 1,000.

► **Example E-10: Property range expression in XML**

```
<property-range name="r_content_size" operator="BETWEEN" repeating="false">
    <from>0</from>
    <to>10000</to>
</property-range>
```



► **Example E-11: Property range expression in JSON**

```
{
    "expression-type": "property-range",
    "name": "r_content_size",
    "operator": "BETWEEN",
    "from": "0",
    "to": "10000",
    "repeating": false
}
```



## E.1.5 Relative date expression

A relative date expression defines constraints on date properties using a relative date. A relative date shows the time difference to the current date, with a minus sign (-) indicating a certain amount of time before the current date, and a plus sign, which can be omitted, indicating a certain amount of time after the current date. The following table describes the details:

**Table E-4: Relative date expression**

Option	Description
name	Time related property name, such as <code>r_modify_date</code>
time unit	Supports these time units: MINUTE, DAY, MONTH, YEAR
value	Number of the time unit
operator	Supports these operators: EQUAL, GREATER_THAN, LESS_THAN, GREATER_EQUAL, LESS_EQUAL

For example, the following expression returns `true` when `r_modify_date` is greater than the current date minus 10 months:

➡ **Example E-12: Relative date expression in XML**

```
<relative-date name="r_modify_date" operator="GREATER_THAN"
    time-unit="MONTH">-10</relative-date>
```



➡ **Example E-13: Relative date expression in JSON**

```
{
  "expression-type": "relative-date",
  "name": "r_modify_date",
  "value": -10,
  "time-unit": "MONTH",
  "operator": "GREATER_THAN",
  "repeating": false
}
```



### E.1.5.1 Expressions as templates

Expressions in AQL can also be used to define search templates. When the attribute “template” is *true*, specific elements in the expressions are activated as variables.

The following table lists all AQL expressions that support variables:

**Table E-5: Expressions as templates**

Expression type	Variable	Sample (variable predicates highlighted in bold)
fulltext	Fulltext search criterion	<fulltext fuzzy="false" template="true"> <b>key word</b> </fulltext>
property	Property value  Note: The search operator cannot be IS_NULL or IS_NOT_NULL.	<property name="object_name" operator="CONTAINS" exact-match="false" repeating="false" case-sensitive="false" fuzzy="false" template="true"> <b>myobj</b> </property>
property list	Value list	<property-list name="r_object_type" operator="IN" repeating="false" template="true"> <values> <value> <b>dm_document</b> </value> <value> <b>dm_folder</b> </value> </values> </property-list>
relative date	Relative value and time unit	<relative-date name="r_modify_date" time-unit="MONTH" operator="GREATER_THAN" repeating="false" template="true">-3</relative-date>
property range	From and To values	<property-range name="r_modify_date" operator="BETWEEN" repeating="false" template="true"> <from>2010-10-13T07:33:13.009+00:00</from> <to>2014-11-11T01:35:17.089+00:00</to> </property-range>

## E.2 Facet in AQL

Using AQL, a facet definition becomes more powerful. You can define multiple facets in one query document. These facets can also be nested. Furthermore, facet grouping strategy and facet ordering can both be configured in AQL.

## E.2.1 Grouping strategy

A grouping strategy defines how to group the results or a facet search. Facet definitions in AQL support the following grouping strategies:

- string
- range
- alpharange
- date

### ➡ Example E-14: String grouping strategy in XML

```
<facet-definition id="id1" group-by="string">
  <attributes>
    <attribute>r_object_type</attribute>
  </attributes>
</facet-definition>
```



### ➡ Example E-15: String grouping strategy in JSON

```
{
  "id": "id1",
  "attributes": [
    "r_object_type"
  ],
  "group-by": "alpharange",
}
```



The `string` grouping strategy is commonly used when a string property is set as the facet, for example, `r_object_type`. This grouping strategy has the following options:

- `nullValueFacet` (optional): indicates whether or not to take null values into account when computing the facet grouping, which defaults to `false`.
- `keepDuplicateValues` (optional): indicates whether or not to take duplicated values for repeating properties into account when computing the facet grouping, which defaults to `false`. This option works for xPlore 1.1 P03 and later versions.
- `caseInsensitive` (optional): indicates whether or not to take letter case into account when computing the facet grouping, which defaults to `false`. This option works for xPlore 1.1 P03 and later versions.

### ➡ Example E-16: Range grouping strategy in XML

```
<facet-definition id="id1" group-by="range">
  <attributes>
```

```

<attribute>r_content_size</attribute>
</attributes>
<properties>
  <property name="range">0:1000,1000:10000,10000:</property>
</properties>
<sort>FREQUENCY</sort>
</facet-definition>

```



#### ➤ Example E-17: Range grouping strategy in JSON

```

{
  "id": "id1",
  "attributes": [
    "r_content_size"
  ],
  "group-by": "range",
  "sort": "FREQUENCY",
  "properties": {
    "range": "0:1000,1000:10000,10000:"
  }
}

```



The range grouping strategy is commonly used when a numeric property is set as the facet, for example, r\_content\_size. This grouping strategy has the following option:

- range (required): Specifies a list of buckets for grouping. Each bucket is separated by a comma (','). A bucket is composed of a lower and upper bounds separated by a colon (':'). You can create open ended buckets by omitting one of the bounds but keeping the comma separator. Overlapping buckets are not supported and may result in inconsistent results.



#### Notes

- The lower-bound is inclusive while the upper-bound is exclusive.
- The boundaries are parsed as Double. Example: 0:10,10:100.89,100:y

#### ➤ Example E-18: Alpharange grouping strategy in XML

```

<facet-definition id="id1" group-by="alpharange">
  <attributes>
    <attribute>r_object_name</attribute>
  </attributes>
  <properties>
    <property name="range">a:m,n:r,s:z</property>
  </properties>
  <sort>FREQUENCY</sort>
</facet-definition>

```



 **Example E-19: Alpharange grouping strategy in JSON**

```
{  
    "id": "id1",  
    "attributes": [  
        "r_object_name"  
    ],  
    "group-by": "alpharange",  
    "sort": "FREQUENCY",  
    "properties": {  
        "range": "a:m,n:r,s:z"  
    }  
}
```



The alpharange grouping strategy is commonly used when a string property is set as the facet, for example, r\_object\_name. This grouping strategy has the following option:

range (required): specifies a list of buckets for grouping. Each bucket is separated by comma(','). A bucket is composed of a lower and upper bounds separated by colon(':'). You can create open ended buckets by omitting one of the bounds but keeping the comma separator. Overlapping buckets are not supported and may result in inconsistent results.

**Notes**

- This grouping strategy is case-insensitive.
- The boundaries are inclusive.
- The boundaries are parsed as characters. Example: a:m,n:r,s:z

 **Example E-20: Date grouping strategy in XML**

```
<facet-definition id="id1" group-by="week">  
    <attributes>  
        <attribute>r_modify_date</attribute>  
    </attributes>  
    <sort>FREQUENCY</sort>  
</facet-definition>
```

 **Example E-21: Date grouping strategy in JSON**

```
{  
    "id": "id1",  
    "attributes": [  
        "r_modify_date"  
    ],  
    "group-by": "week",  
}
```

```
    "sort": "FREQUENCY"  
}
```



The date grouping strategy has the following types:

- day: groups results by day
- week: groups results by week
- month: groups results by month
- quarter (since xPlore 1.3): groups results by quarter
- year: groups results by year
- relativeDate: a pre-defined grouping, which contains the following groups:  
OLDER, LAST\_YEAR, LAST\_MONTH, LAST\_WEEK, THIS\_YEAR, THIS\_WEEK,  
YESTERDAY, TODAY

For more information on the options for this grouping strategy, see the [Grouping Strategies table on page 662](#).

## E.2.2 Facet ordering

Facet ordering is configured in the `sort` element of a facet definition. The following ordering policies are supported:

- FREQUENCY: sort by the result count in descending order.
- NONE: keeps the order returned by the facet handler.
- VALUE\_ASCENDING: sort by the facet value ID in ascending order.
- VALUE\_DESCENDING: sort by the facet value ID in descending order.



## Appendix F. Events Audited by Default

Some events are audited by default in Documentum CM Server. The following is a listing of the events that are audited by default. In each point we list the event (or events) and a short description:

- `dm_audit` and `dm_unaudit`  
All executions of methods that register or unregister events for auditing.
  - `dm_signoff`  
All executions of a `<IDfPersistentObject.signoff>` method.
  - `dm_adddigisignature`  
All executions of an `<addDigitalSignature>` method. By default, audit trail entries created for the `<dm_adddigisignature>` event are not signed by Documentum CM Server. To sign those entries, register the event explicitly for auditing, then set the argument to require signing.
  - `dm_addesignature` and `dm_addesignature_failed`  
All executions, successful or unsuccessful, of an `<addESignature>` method. The audit trail entries for the `<dm_addesignature>` event are automatically signed by Documentum CM Server.
  - `dm_purgeaudit`  
Removal of an audit trail entry from the repository. A `<dm_purgeaudit>` event is generated whenever a destroy method is executed to remove an Audit Trail entry from the repository, or a `<PURGE_AUDIT>` administration method is executed. All `<dm_purgeaudit>` events are audited.
  - User login failure
  - `dm_default_set`  
A default set of events for objects of type `<dm_document>` and its subtypes. This event is registered against the docbase configuration object, and Documentum CM Server audits the events in the set for `<dm_document>` type and its subtypes.
- The following events are registered against the docbase configuration object:
- `dm_archive`
  - `dm_checkin`
  - `dm_restore`
  - `dm_assemble`
  - `dm_checkout`
  - `dm_save`
  - `dm_bp_attach`
  - `dm_destroy`

- dm\_setfile
- dm\_bp\_demote
- dm\_freeze
- dm\_signoff
- dm\_bp\_promote
- dm\_link
- dm\_unfreeze
- dm\_bp\_resume
- dm\_lock
- dm\_unlink
- dm\_bp\_suspend
- dm\_mark
- dm\_unlock
- dm\_branch
- dm\_prune