

**DCAT Concept Document**

|  |  |
| --- | --- |
| Team Name: | HCL |
| Created By: | HCL Team |

Table of Contents

[1.0 Revision History 3](#_Toc370397918)

[2.0 Major Frameworks and Plug-In Used 3](#_Toc370397919)

[3.0 DCAT-DCTM Integration 3](#_Toc370397920)

[3.1 DCAT Metadata Service 3](#_Toc370397921)

[3.1.1 Code description 3](#_Toc370397922)

[3.2 DCAT Content Service 8](#_Toc370397923)

[3.2.1 GetContentServlet 8](#_Toc370397924)

[3.2.2 DQL Used 8](#_Toc370397925)

[4.0 DCAT Asset DAO 8](#_Toc370397926)

[4.1 DB DAO 8](#_Toc370397927)

[4.2 DCTM DAO 9](#_Toc370397928)

[5.0 DCTM Asset Download with Navigation 9](#_Toc370397929)

[6.0 DCTM Asset Download with Placeholder Functionality 10](#_Toc370397930)

[7.0 Bulkdownload Job Functionality 11](#_Toc370397931)

[7.1 Select a Folder from Navigation Pane: 11](#_Toc370397932)

[7.2 Bulkdownload Job 11](#_Toc370397933)

[8.0 DCTM Asset Sync Utility 11](#_Toc370397934)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Name | Date | Comments |
| 1.0 | CMS Team | 26rd Dec, 2012 | Initial Draft |

# Major Frameworks and Plug-In Used

|  |  |
| --- | --- |
| **Framework and Plug-in** | **Version** |
| JDK | 1.6 and Above |
| Play Framework | 1.2.3 |
| Spring | 3.1.2 |
| JQuery | 1.7.1 |
| TinyMCE | 2.1 |
| MathJAX | 2.0 |
| Maven | 2.0 |
| Eclipse | Indigo and above |

# DCAT-DCTM Integration

DCAT is integrated with DCTM for only downloading Assets. There is no other functions are available in DCAT. To integrate with DCTM we have created two rest services:

1. DCAT Metadata Service
2. DCAT Content Service

## DCAT Metadata Service

DCAT Metadata service provides metadata information of DCTM Objects like files and folders. You can get metadata of DCTM Objects by following manners:

1. Metadata for single file
2. Metadata for all objects inside a folder
3. Metadata for only folders inside a folder
4. Metadata for only files inside a folder

### Code description

SVN Location:

<http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/DCATMetadata>

This service is built using “Spring framework” and “Maven”. Following are the important methods:

#### getContentData Function

This method will use to get metadata of objects inside a folder from DCTM. The URL Structure to use this Function is as follows:

http://<host>:<port>/dcatmetadata/dcatservice/objectmetadata?path=<DCTM\_Folder\_Path>&descend=<true/false>&mimetype=<content\_type\_of\_file>&contentclass=<all/file/folder>&docbase\_name=<<docbase\_name>>

e.g. <http://oldtut045.pearsontc.com:50361/dcatmetadata/dcatservice/objectmetadata?path=/DCAT/test1&descend=false&mimetype=&contentclass=file&docbase_name=schcmsdevdoc>

1. **Signature of Method**

|  |
| --- |
| public @ResponseBody ContentDatasSearchResultTp getContentData(@RequestParam("path") String path, @RequestParam("descend") String descend,@RequestParam("mimetype") String mimeType , @RequestParam("contentclass") String contentClass ) |

1. **Request Mapping URL**

|  |
| --- |
| @RequestMapping(value="/objectmetadata",method = RequestMethod.GET) |

1. **Parameters for Method**
2. **path**: - DCTM folder path e.g. /DCAT/test
3. **descend**: - true if need to include subdirectory too else false.
4. **mimetype**: - format option for asset e.g. if you pass “jpg” it will return only jpg files.
5. **contentclass**: - there are three options for content class
   * 1. **all** :- it will return all objects folders as well as files
     2. **folder** :- it will return only folders
     3. **file** :- it will return only files
6. **docbase\_name:-** Documentum repository name
7. **Response**

As a response it returns an XML string. Basic structure of XML string is as follows:

|  |
| --- |
| <contentDatasSearchResultTp>  <ResultSet>  </DataRows>  </ResultSet>  </contentDatasSearchResultTp> |

Following are the attributes for DataRows element:

1. **IsFolder**: - true if object folder. False if object is file.
2. **ModificationData**: - Object Modification date in DCTM
3. **GUID**: - DCTM GUID for Object

Following are the sub elements from the DataRows element:

1. **ChronicleID**: - value of i\_chronicle\_id attribute of DCTM object. Null for Folders.
2. **ContentClass**: - value of prsn\_content\_class attribute of DCTM object. Null for Folders.
3. **ContentClassTier1**: - value of prsn\_content\_type\_tier1 attribute of DCTM object. Null for Folders.
4. **ContentClassTier2**: - value of prsn\_content\_type\_tier2 attribute of DCTM object. Null for Folders.
5. **ContentID**: - value of i\_contents\_id attribute of DCTM object. Null for Folders.
6. **Description**: - value of prsn\_description attribute of DCTM object.
7. **Keywords**: - value of keywords attribute of DCTM object separated with “|”.
8. **Name**: - value of object\_name attribute of DCTM object.
9. **SourceURI**: - URL for DCAT Content Download service.
10. **ThumbnailURI**: - value of thumbnail\_url attribute of DCTM object.

#### getContentDataByName Function

This method will use to get metadata of a single file from DCTM. The URL Structure to use this Function is as follows:

http://<host>:<port>/dcatmetadata/dcatservice/filemetadata?path=<DCTM\_FOLDER\_PATH>&filename=<DCTM\_FILE\_PATH>

e.g. <http://oldtut045.pearsontc.com:9000/dcatmetadata/dcatservice/filemetadata?path=/DCAT/test1&filename=a%201.jpg>

1. **Signature of Method**

|  |
| --- |
| public @ResponseBody ContentDatasSearchResultTp getContentDataByName(@RequestParam("path") String path, @RequestParam("filename") String name ) |

1. **Request Mapping URL**

|  |
| --- |
| @RequestMapping(value="/filemetadata",method = RequestMethod.GET) |

1. **Parameters for Method**
2. **path**: - DCTM folder path e.g. /DCAT/test
3. **filename**: - DCTM File name. e.g. test1.jpg
4. **Response**

As a response it returns an XML string. Basic structure of XML string is as follows:

|  |
| --- |
| <contentDatasSearchResultTp>  <ResultSet>  </DataRows>  </ResultSet>  </contentDatasSearchResultTp> |

Following are the attributes for DataRows element:

1. **IsFolder**: - always false.
2. **ModificationData**: - Object Modification date in DCTM
3. **GUID**: - DCTM GUID for Object

Following are the sub elements from the DataRows element:

1. **ChronicleID**: - value of i\_chronicle\_id attribute of DCTM object. Null for Folders.
2. **ContentClass**: - value of prsn\_content\_class attribute of DCTM object. Null for Folders.
3. **ContentClassTier1**: - value of prsn\_content\_type\_tier1 attribute of DCTM object. Null for Folders.
4. **ContentClassTier2**: - value of prsn\_content\_type\_tier2 attribute of DCTM object. Null for Folders.
5. **ContentID**: - value of i\_contents\_id attribute of DCTM object. Null for Folders.
6. **Description**: - value of prsn\_description attribute of DCTM object.
7. **Keywords**: - value of keywords attribute of DCTM object separated with “|”.
8. **Name**: - value of object\_name attribute of DCTM object.
9. **SourceURI**: - URL for DCAT Content Download service.
10. **ThumbnailURI**: - value of thumbnail\_url attribute of DCTM object.

#### DQL Used

1. **All Contents Query: -** This query is used to get metadata for all files inside a folder path from DCTM.

|  |
| --- |
| select r\_object\_id,object\_name,prsn\_content\_class, prsn\_content\_type\_tier1,prsn\_content\_type\_tier2, prsn\_description, keywords, thumbnail\_url, object\_name, prsn\_guid, r\_modify\_date, i\_folder\_id, i\_contents\_id from prsn\_content where FOLDER(<<FOLDER\_PATH>>) and a\_content\_type != 'gif' |

1. **Folder Query: -** This query is used to get metadata for all folders inside a folder path from DCTM.

|  |
| --- |
| select r\_object\_id,object\_name,prsn\_description, keywords, thumbnail\_url, object\_name, prsn\_guid, r\_modify\_date, i\_folder\_id, i\_contents\_id from prsn\_general\_folder where FOLDER(<<FOLDER\_PATH>>) UNION select r\_object\_id,object\_name,prsn\_description, keywords, thumbnail\_url, object\_name, prsn\_guid, r\_modify\_date, i\_folder\_id, i\_contents\_id from prsn\_folder where FOLDER(<<FOLDER\_PATH>>) UNION select r\_object\_id,object\_name,prsn\_description, keywords, thumbnail\_url, object\_name, prsn\_guid, r\_modify\_date, i\_folder\_id, i\_contents\_id from prsn\_common\_folder where FOLDER(<<FOLDER\_PATH>>) UNION select r\_object\_id,object\_name,prsn\_description, keywords, thumbnail\_url, object\_name, prsn\_guid, r\_modify\_date, i\_folder\_id, i\_contents\_id from prsn\_product\_folder where FOLDER(<<FOLDER\_PATH>>) |

1. **Format Query: -** This query is used to get all formats available in DCTM.

|  |
| --- |
| select distinct name from dm\_format where mime\_type in (<<mime\_types>>) |

1. **Folder Path Query: -** this query is used to get folder path of a folder from DCTM.

|  |
| --- |
| select r\_object\_id, r\_folder\_path from dm\_folder where any r\_folder\_path like '<<FOLDER\_PATH>>' |

1. **Single Content Query: -** This query is used to get metadata for single file inside a folder path from DCTM.

|  |
| --- |
| select r\_object\_id,object\_name,prsn\_content\_class, prsn\_content\_type\_tier1,prsn\_content\_type\_tier2, prsn\_description, keywords, thumbnail\_url, object\_name, prsn\_guid, r\_modify\_date, i\_folder\_id, i\_contents\_id from prsn\_content where FOLDER('<<FOLDER\_PATH>>') and object\_name='<<NAME>>' and a\_content\_type != 'gif' |

## DCAT Content Service

SVN Location:

<http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/DCATContent>

This service is built using “JAVA Servlet” and “Maven”.

### GetContentServlet

This servlet will use to download the content from DCTM. It is a simple JAVA Servlet.

* + - 1. **Servlet Class: -** com.pearson.dcat.dctm.object.util.GetContentServlet
      2. **URL Pattern: -** /content/\*
      3. **Parameters: -**

1. objecteId: - r\_object\_id of a DCTM File.
   * + 1. **Response: -** it gets the object id from request. Read the properties file to get the Documentum repository session information and create Documentum session. Then it get the object by object id and return as OutputStream object as response.

### DQL Used

* + - 1. **GET DCTM Object: -** dm\_document where r\_object\_id='"+objectID+"'";

# DCAT Asset DAO

We have created DAO layer to contact with DCTM and DB.

It is spring project to connect with database to run queries and also contact with DCTM related service like DCTM Content and DCTM Metadata service.

SVN Location:

<http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/DCATAssetDAO>

## DB DAO

Currently it support of Query on following tables only:

1. BULKDOWNLOADJOB

|  |  |
| --- | --- |
| Interface Name: | IBulkdownloadJob.java |
| Class Name: | BulkdownloadDao.java |
| Method available: |  |
|  | **void** setDataSource(DataSource ds) |
|  | List<BulkdownloadJob> select(HashMap<String, Object> filterMap) |
|  | List<BulkdownloadJob> selectAll() |
|  | **void** deleteAll() |
|  | **void** delete(**long** id) |
|  | **void** create(BulkdownloadJob dctmAsset) |
|  | **void** update(BulkdownloadJob dctmAsset) |
|  | **void** logError(ErrorLog error) |
|  | List<BulkdownloadJob> selectByJobId(Long jobId) |
|  | **void** setJobStatus(Long jobId, String status,Date endTime) |
|  | **void** unLockSCO(String scoId) |

1. DCTM\_ASSET\_DETAILS

|  |  |
| --- | --- |
| Interface Name: | IDCTMAssetDao.java |
| Class Name: | DCTMAssetDao.java |
| Method available: |  |
|  | **void** setDataSource(DataSource ds) |
|  | List<DCTMAssetDetails> select(HashMap<String, Object> filterMap) |
|  | List<DCTMAssetDetails> selectByPlaceholder(boolean flag ,String instanceKey) |
|  | List<DCTMAssetDetails> selectAll() |
|  | void deleteAll() |
|  | void delete(long id) |
|  | void create(DCTMAssetDetails dctmAsset) |
|  | void update(DCTMAssetDetails dctmAsset) |
|  | void logError(ErrorLog error) |
|  | void unlock(String instanceKey) |
|  | List<DCTMAssetDetails> getReadytoSyncAssets() |

## DCTM DAO

Currently it support only following Service of DCTM:

1. DCTM Metadata Service
2. DCTM Content Service
3. Bulkdownload Job

# DCTM Asset Download with Navigation

There are various ways to download Assets from Documentum in DCAT. One of them is downloading via “Select From Documentum Tab”. To use this feature you need to do following steps:

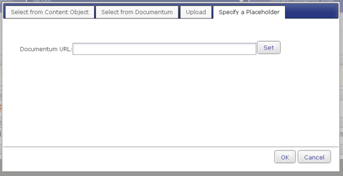
* 1. Login in DCAT.
  2. Select Program
  3. Go to Home page in DCAT
  4. Edit content of any type of SCO
  5. Edit any Page and open Authoring Form for any gadget which has any asset.
  6. Choose any type if asset in Authoring form and open Asset Gadget Window.
  7. Click on the “Select From Documentum Tab”
  8. There would be two panes visible.
  9. One is for Folder selection and other is for Asset Selection
  10. User can navigate from Folder Pane and download asset from Content Pane.

We use DCAT Metadata Service to get the Metadata of Folders and Content and display it on the Panes. When a User click on and Asset it call the DCAT Content Service and download the files.

# DCTM Asset Download with Placeholder Functionality

If an asset is not yet available in DCTM, User can set a Place holder from “Set Placeholder” Tab.

Here you can put placeholder value and click on “Set” button and click on “ok” button. The placeholder will hit the Documentum to check if the asset exists in Documentum or not. If it exists then download the asset to the correct directory and update the database table. If, it does not exist then make it makes entry in database and saves image as “Image Coming Soon”. Later it gets replaced when image comes in Documentum.



There is a two part of Placeholder functionality:

* + - 1. **Put Placeholder data inside database:**

There are three ways by which user can set a placeholder

1. Only Asset file name
2. Full DCTM Path
3. Combination of SCO ID / Asset Name

User can put asset details as mentioned above and click on the Set Button. DCAT will save details of Asset inside the Database which will be access by Placeholder Job.

* + - 1. **Placeholder Job:**

A spring based Job created for Placeholder Assets. It runs in every 15 min and gets the assets from Database and check for DCTM Availability. If the asset is available it will download else it will try in next run.

SVN Location:

[**http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/AssetPlaceholderJob**](http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/AssetPlaceholderJob)

# Bulkdownload Job Functionality

In DCAT, user can download a complete folder by Bulkdownload Job functionality.

There is two part of this functionality:

## Select a Folder from Navigation Pane:

In DCAT user can open a Navigation Pane from Gadget Pane and select the folder path. While clicking Update button DCAT create a JMS Request to Active MQ and save the Details in database.

## Bulkdownload Job

There is a Spring based Job which is linked to Active MQ JMS. It continuously pings the Active MQ and when the new Job comes it get the Data from Active MQ and Database and download the complete folder inside DCAT SCO.

SVN Location:

<http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/BulkdownloadConsumer>

# DCTM Asset Sync Utility

Once the Asset is downloaded inside SCO folder and if user changes in Documentum, DCAT has a DCTM Asset Sync Utility.

This utility runs once in a day and gets all the asset detail from Database and check for new version in DCTM.

Every time when some content changes inside the DCTM the i\_contents\_id change. DCAT saves DCTM Content ID in database every time asset downloads. This Utility checks the current version Content ID and match with the saved Content ID. If the content id is change then it download new version of asset.

SVN Location:

<http://subversion.pearsoncmg.com/data/ecms/DCAT-DCTM/trunk/DCTMAssetSyncUtility>