## DME Design of including remote file system path in the return values from Data Management Model API

Design Document

Version 1.0

8/07/2025

# Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version Number | Implemented By | Revision Date | Approved By | Approval Date | Description of Change |
| 1.0 | Sarada Chintala |  |  |  | Initial Draft |

Table of Contents

[DME Design of including remote file system path in the return values from Data Management Model API 1](#_Toc205476458)

[Version History 2](#_Toc205476459)

[Introduction 4](#_Toc205476460)

[Purpose of this document 4](#_Toc205476461)

[Software Design 4](#_Toc205476462)

[High Level Design 4](#_Toc205476463)

[Detailed Design 5](#_Toc205476464)

[Data Structure Change: 5](#_Toc205476465)

[Populating the new field in the DataManagement Model API 6](#_Toc205476466)

[Test Data: 6](#_Toc205476467)

# Introduction

## Purpose of this document

The purpose of this document is to record the detailed design of including remote file system path in the return values from Data Management Model API.

# Software Design

## High Level Design

The current DataManagementModel structure returns a collection of ***docRules***. Each entry in the ***docRules*** contains a ***doc*** and a multiple set of ***rules*** associated with each ***basePath***. As shown below each instance of ***rules*** has an associated ***basePath***.

{

"docRules": [

{

"doc":"TESTDOC",

"rules": [{

"id":"f1b5a658-610d-4c34-a211-bc19799a4a6d",

"basePath": "/TEST\_Archive",

"collectionMetadataValidationRules": [ …],

"dataObjectMetadataValidationRules": [… ],

"dataHierarchy": {}

}],

"collectionSystemGeneratedMetadataAttributeNames": [],

"dataObjectSystemGeneratedMetadataAttributeNames": [],

}…] …}

The new requirement is that the remote file system path needs to be added to the above model. The remote file system is associated with a ***basePath***. The resulting structure will now contain the new field ***externalArchivePath***:

{

"docRules": [

{

"doc":"TESTDOC",

"rules": [{

"id":"f1b5a658-610d-4c34-a211-bc19799a4a6d",

"basePath": "/TEST\_EXT\_Archive",

"externalArchivePath: '/vast/testDirectory/TEST\_Archive',

"collectionMetadataValidationRules": [ …],

"dataObjectMetadataValidationRules": [… ],

"dataHierarchy": {}

} ….],

"collectionSystemGeneratedMetadataAttributeNames": […],

"dataObjectSystemGeneratedMetadataAttributeNames": […],

}…] }

## Detailed Design

### Data Structure Change:

The HpcDataManagementRulesDTO data structure will have a new field ***externalArchivePat***h as shown below:

<!-- Data Management Model DTO -->

<xsd:complexType name="HpcDataManagementRulesDTO">

<xsd:sequence>

<xsd:element name="id" type="xsd:string" />

<xsd:element name="basePath" type="xsd:string" />

<xsd:element name="externalArchivePath" type="xsd:string" />

<xsd:element name="collectionMetadataValidationRules"

type="hpc-domain-metadata:HpcMetadataValidationRule" minOccurs="0"

maxOccurs="unbounded" />

<xsd:element name="dataObjectMetadataValidationRules"

type="hpc-domain-metadata:HpcMetadataValidationRule" minOccurs="0"

maxOccurs="unbounded" />

<xsd:element name="dataHierarchy"

type="hpc-domain-datamanagement:HpcDataHierarchy" />

</xsd:sequence>

</xsd:complexType>

### 

### Populating the new field in the DataManagement Model API

The following is the pseudo code to update the new field ***externalArchivePath***:

Get all datamanagement configurations

For each datamanagement configuration {

If the ***basePath*** has been provided as a parameter in the API{

Check if the ***basePath*** matches the datamanagement configuration ***basePath***

before building ***rules***

}

Get the associated DOC

/\*\* ***rules*** is the above data structure HpcDataManagementRulesDTO \*\*/

build ***rules*** {

Get the ***basePath*** from the DataManagement Configuration

Get the S3Configuration using the S3UploadConfiguration

If( ( ***S3Configuration.externalStorage*** is true

&& ***S3Configuration.POSIX\_PATH*** is populated) {

***rules***.***externalArchivePath = s3Configuration.posix\_path***;

}

}

}

### Test Data for DEV and UAT:

The following fields in the Config structures are relevant to testing this feature.

BASEPATH: TEST\_EXT\_Archive

DOC: DME\_TEAM

External Directory: /mnt/DMETemp/test\_auto\_tiering/

The external\_storage and posix\_path fields need to be populated in the S3 configuration structure:

HPC\_S3\_ARCHIVE\_CONFIGURATION

EXTERNAL STORAGE : 1

POSIX\_PATH: : /mnt/DMETemp/test\_auto\_tiering/

In the dataManagement configuration, GLOBUS\_ARCHIVE\_TYPE needs to be set to TEMPORARY\_ARCHIVE for POSIX.

HPC\_DATA\_MANAGEMENT\_CONFIGURATION

GLOBUS\_ARCHIVE\_TYPE: TEMPORARY\_ARCHIVE