

Validation only for selected resources

Title (Goal)	Validation only for selected resources
Primary Actor	Information architect, developer
Scope	Component
Level	Summary
Author	Stefano Cossu
Story (A paragraph or two describing what happens)	Enable mandatory and/or optional content and structural validation only for certain resources

This use case exemplifies a repository in which validation can be applied to arbitrary resources instead of a whole repository. This validation can consist of mandatory and/or recommended rules, as described in [Enforce validation across repository](#) and [Optional validation](#).

Following a discussion on this wiki [1], a hierarchy has been considered a not recommended way to group resources that should or should not be validated. Containment predicates such as ldp:contains or pcdm:hasMember, or assigning RDF types to individual resources to be validated, is a better choice.

Example 1: validate resources under a specific container

1. User uploads an image of type cma:Image under /container_a/image1
2. User uploads an image of type cma:Image under /container_b/image2
3. User uploads an image of types cma:Image and cma:Validatable under /container_b/image3
4. /container_a is of type cma:Validatable
5. /container_b is not cma:Validatable
6. Validation rules are defined for cma:Image in a configuration
7. Configuration also indicates that rules should only be checked if the resource is of type cma:Validatable, or is contained by a container of that type
8. Validation **is** checked for /container_a/image1 since it is under a cma:validatable container (i.e. </container_a> ldp:contains </container_a /image1>)
9. Validation **is not** checked for /container_b/image2
10. Validation **is** checked for /container_b/image3 because this resource itself is cma:validatable

Example 2: Indicate conditions for individual rules

A validation configuration specifies that:

1. Property myns:uid:
 - a. is mandatory
2. Property myns:documentType:
 - a. is mandatory
 - b. must be checked if the resource is of type cma:validatable or is in a container of such type
3. Property myns:creator:
 - a. is mandatory
 - b. must be checked if the resource is of type cma:validatable or is in a container of such type
 - c. if validation fails, request should be forwarded to a specified service which forwards the request to Fedora and issues a warning that this property should be present

Given the resources being ingested in Example 1:

1. for all three resources, if myns:uid is missing, ingest fails
2. for /container_a/image1 and container_b/image3:
 - a. if myns:documentType is missing, ingest fails
 - b. if myns:creator is missing, ingest proceeds and a warning is issued

[1] [Re: AIC Use Case: Content and Structural Validation](#)