OSID Specification Request: 8
Track: Interface
Status: Accepted

Tom Coppeto
Ingenescus

June 2014

# Catalog Assignment

## Status

This document is a request for a specification change for review.

## Summary

OsidObjects can be manually assigned and unassigned from OsidCatalogs. This document proposes adding a new operation to atomically reassign an OsidObject from one OsidCatalog to another.

## Table of Contents

## 1. Current Specification

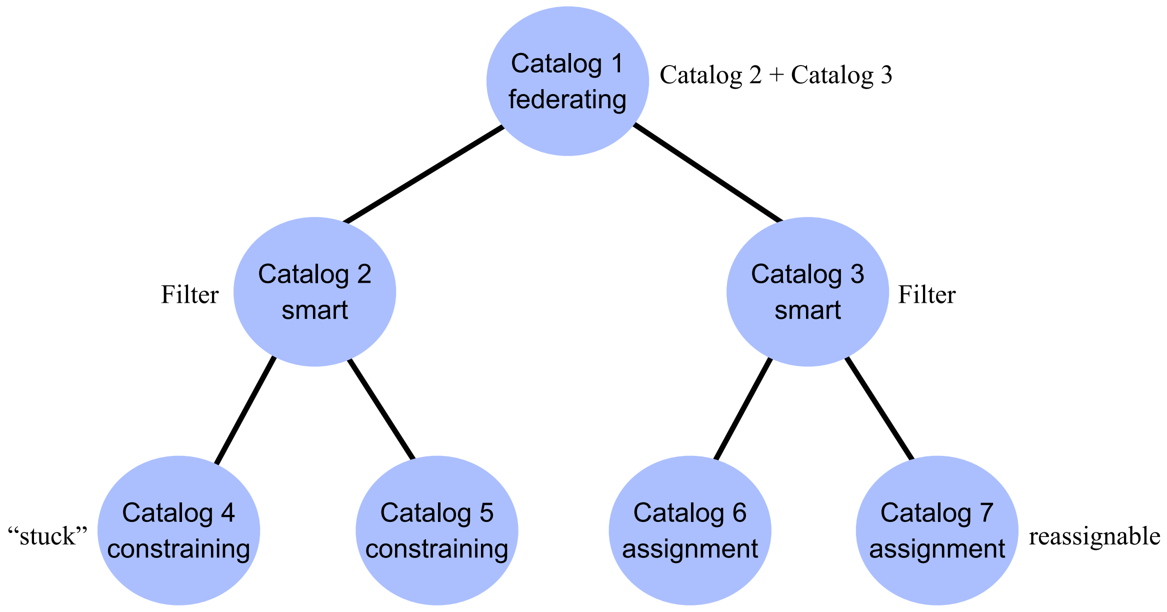
OsidObjects are created in the context of an OsidCatalog. The OsidObject may or may not be permanently attached to the OsidCatalog in which it was created. An OsidObject is never restricted from appearing in other OsidCatalogs, however an OsidCatalog may restrict the set of OsidObjects that may appear in it.

The appearance of OsidObjects in OsidCatalogs is the result of one of the following scenarios:

1. The OsidObject was created in the OsidCatalog using an administrative OsidSession.
2. The OsidObject was assigned to the OsidCatalog using a catalog assignment OsidSession.
3. The OsidObject is the result of an OsidQuery attached to the OsidCatalog.
4. The OsidCatalog is a federating parent of another OsidCatalog that reveals the OsidObject through one of the above scenarios.

Catalog assignment OsidSessions provide a means to manually assign and unassign OsidObjects to OsidCatalogs. An OSID Provider supporting this service does not guarantee that any OsidObject may be assigned to, or unassigned from, any OsidCatalog. Constraints may exist which determine success.

OsidObjects must appear in at least one OsidCatalog because OsidObjects can only be accessed through an OsidCatalog. While the specification states that removing an OsidObject from the last OsidCatalog is the equivalent of deleting it, an OSID Provider may elect to fail the operation. An OSID Provider may also decide to reassign the removed OsidObject to another OsidCatalog (e.g. Trashcan).



OsidCatalogs in a hierarchy may exhibit different behaviors or a combination of behaviors. For example, an OsidCatalog which has a filter may also be used as a constraint for the kinds of OsidObjects that may be created in it. A federating OsidCatalog may also have additional OsidObjects explicitly assigned to it. The possibilities are complex and not explicitly spelled out in the specification.

There are hints to guide an OSID Consumer. First is the ability to access either the admin, smart catalog, or catalog assignment OsidSessions using a specific OsidCatalog. OSID Providers may deny access outright depending on the kind of OsidCatalog it is managing. It may also opt to permit the OsidSession but indicate no access using the pre-authorization tests.

## 2. Problem

There is not a means of reassigning an OsidObject to another OsidCatalog in a single operation.

## 3. Proposed Changes

Add a method to reassign an OsidObject.

## 4. Impacts

### 4.1. Specification

This change requires adding a method to all assignment OsidSessions.

### 4.2. OSID Consumers

OSID Consumers can ignore the new method.

### 4.3. OSID Providers

OSID Providers would be expected to support the reassignment operation. Although it can be a simple combination of assign and unassign, the implied transactionality should be factored into the implementation.

## 5. Interoperability Considerations

The interoperability of OsidCatalog assignment can be characterized by:

- support for assigning to the specified OsidCatalog
- support for unassigning from the specified OsidCatalog

Although these operations are *mandatory*, an OSID Provider may not allow assignment or unassignment of any or all OsidObjects to and from any or all OsidCatalogs. A reassignment operation introduces a third interoperability concern in whether or not an OSID Provider can support an atomic move from one OsidCatalog to another.

Moreover, the concern that an OsidObject can reside in multiple OsidCatalogs is never addressed. It may be possible that an assign/unassign sequence of operations fails because an OSID Provider requires it to live in one OsidCatalog or another. Another interpretation of the assign operation is to implicitly perform an unassign operation thus making it a reassign operation.

This proposal specifies an explicit operation to split the reassign operation from the assign operation thus clarifying the usage of these methods.

## 6. Proposed Interfaces

### 6.1. Example Assignment OsidSession

| ***Interface*** | **osid.messaging.MessageMailboxAssignmentSession** | | |
|---|---|---|---|
| **Implements** | osid.OsidSession | | |
| **Description** | This session provides methods to re-assign Messages to Mailboxes. A Mailbox may map to multiple Mailboxes and removing the last reference to a Message is the equivalent of deleting it. Each Mailbox may have its own authorizations governing who is allowed to operate on it. | | |
| **Method** | **canAssignMessages** | | |
| **Description** | Tests is this user can alter message/mailbox mappings. A return of true does not guarantee successful authorization. A return of false indicates that it is known mapping methods in this session will result in a PERMISSION_DENIED. This is intended as a hint to an application that may opt not to offer lookup operations to unauthorized users. | | |
| **Return** | false | | false if mapping is not authorized, true otherwise |
| **Compliance** | mandatory | | This method must be implemented. |
| **Method** | **canAssignMessagesToMailbox** | | |
| **Description** | Tests is this user can alter message/mailbox mappings. A return of true does not guarantee | | |
| **Parameters** | osid.id.Id | mailboxId | the Id of the mailbox |
| **Return** | false | | false if mapping is not authorized, true otherwise |
| **Errors** | NULL_ARGUMENT | | mailboxId is null |
| **Compliance** | mandatory | | This method must be implemented. |
| **Method** | **getAssignableMailboxIds** | | |
| **Description** | Gets a list of mailboxes including and under the given mailbox node in which any message can be assigned. | | |
| **Parameters** | osid.id.Id | mailboxId | the Id of the mailbox |
| **Return** | osid.id.IdList | | list of assignable mailbox Ids |
| **Errors** | NULL_ARGUMENT | | mailboxId is null |
| | OPERATION_FAILED | | unable to complete request |
| **Compliance** | mandatory | | This method must be implemented. |
| **Method** | **getAssignableMailboxIdsForMessage** | | |
| **Description** | Gets a list of mailboxes including and under the given mailbox node in which a specific message can be assigned. | | |
| **Parameters** | osid.id.Id | mailboxId | the Id of the Mailbox |
| | osid.id.Id | messageId | the Id of the Message |
| **Return** | osid.id.IdList | | list of assignable mailbox Ids |
| **Errors** | NULL_ARGUMENT | | mailboxId or messageId is null |
| | OPERATION_FAILED | | unable to complete request |
| **Compliance** | mandatory | | This method must be implemented. |

| | | | |
|---|---|---|---|
| **Method** | **assignMessageToMailbox** | | |
| **Description** | Adds an existing Message to a Mailbox. | | |
| **Parameters** | osid.id.Id | messageId | the Id of the Message |
| | osid.id.Id | mailboxId | the Id of the Mailbox |
| **Errors** | ALREADY_EXISTS | | messageId already assigned to mailboxId |
| | NOT_FOUND | | messageId or mailboxId not found |
| | NULL_ARGUMENT | | mailboxId or messageId is null |
| | OPERATION_FAILED | | unable to complete request |
| | PERMISSION_DENIED | | authorization failure |
| **Compliance** | mandatory | | This method must be implemented. |
| **Method** | **unassignMessageToMailbox** | | |
| **Description** | Removes a Message from a Mailbox. | | |
| **Parameters** | osid.id.Id | messageId | the Id of the Message |
| | osid.id.Id | mailboxId | the Id of the Mailbox |
| **Errors** | NOT_FOUND | | messageId or mailboxId not found or messageId not mapped to mailboxId |
| | NULL_ARGUMENT | | mailboxId or messageId is null |
| | OPERATION_FAILED | | unable to complete request |
| | PERMISSION_DENIED | | authorization failure |
| **Compliance** | mandatory | | This method must be implemented. |
| **Method** | **reassignMessageToMailbox** | | |
| **Description** | Removes aMessage from a Mailbox. | | |
| **Parameters** | osid.id.Id | messageId | the Id of the Message |
| | osid.id.Id | fromMailboxId | the Id of the current Mailbox |
| | osid.id.Id | toMailboxId | the Id of the destination Mailbox |
| **Errors** | ALREADY_EXISTS | | messageId already assigned to mailboxId |
| | NOT_FOUND | | messageId, fromMailboxId, or toMailboxId not found or messageId not mapped to fromMailboxId |
| | NULL_ARGUMENT | | messageId, fromMailboxId, or toMailboxId is null |
| | OPERATION_FAILED | | unable to complete request |
| | PERMISSION_DENIED | | authorization failure |
| **Compliance** | mandatory | | This method must be implemented. |

## 7. Statement