Tom Coppeto
Ingenescus

January 2014

Id Assignment

Status

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

Summary

This document is a request to expand the functionality of the Id OSID to include external management of Ids, creation of Ids based on sequence numbers, and management of external codes.

Table of Contents

1. Current Specification

The IdAdminSession defines a createId() method to return a new Id. This was intended for use as an underlying supporting service for the creation of entities in other OSIDs.

Ids are opaque to parties other than the one responsible for its construction. As an OsidPrimitive, Ids may be constructed by OSID Consumers or OSID Providers. However, only OSID Providers may assign the primary Id in a create operation.

The Id OSID has followed the principle of ownership by OSID Providers in that only the Id OSID Provider may assign an Id. This prevents the Id OSID from being used to manage Ids owned and generated, or partially specified, by its consumer.

The current specification also states that once an Id is assigned to an OsidObject, it is permanent and the Id may never be deleted. The OSIDs live in a fantasy world where entities exist forever and no system change should ever introduce incompatibilities. In other words, it leaves these issues to be addressed outside the specification.

2. Problem

1. Federation

There is an assumption that all Ids are being assigned out of the same authority and namespace. Different OSID ID Providers are required to manage Id assignment for more than one entity.

2. Code Management

It is desirable to be able to register an externally generated Id in the Id OSID to relating internally generated Ids to external codes using the aliasing mechanism.

3. Sequencing

It is desirable to use the Id OSID to manage assignment of unique sequences or portions of an Id identifier for code generation.

4. Conventions

The create operations are well established. Using OsidForms, they communicate what an OSID Consumer may affect and what may be constrained by an OSID Provider. The current createId() method does not follow these conventions.

The batch version of createId is embedded in the same OSID Package where in all other OSIDs it is in its own package.

3. Proposed Change

a. IdIssueSession

This IdIssueSession will handle the simple issuing of new Ids where the OSID Consumer has no control of the Ids created. It will be set apart from the standard create pattern found in the administrative OsidSessions.

b. IdAdminSession

The IdAdminSession will follow standard patterns and create Ids via an IdForm submission.

c. IdForm

The IdForm will allow the components of the Id to be set by its OSID Consumer. It will also be possible to communicate an identifier prefix by which an OSID Provider can manage unique sequence numbers.

d. Id Batch OSID

The batch operations for both issuing and creating Ids will be moved out of the top-level package to the Id Batch OSID.

4. Scope of Change

1. This change does not challenge the "Ids should never go away" doctrine. There is no delete or update operation. The aliasing mechanism in the IdAdminSession requires both Ids be already known to the Id OSID Provider. As such, aliases may be removed and later assigned to other Ids without removing the Ids themselves.

2. The Id remains an OsidPrimitive. The scope of the Id OSID is to define the tools for handling Id assignment and mapping. The Id OSID, like the Type OSID, is not a registry where other information concerning an administrative process can be managed as part of the Id. Such management requires a complete OsidObject.

3. Impacts

3.1. Specification

The set of changes is contained within the Id OSID. Relocation of bulk and issue functions and the addition of a new OsidSession and OsidForm.

3.2. OSID Consumers

Issuing Ids has moved to the IdIssueSession and batch operations moved to a sub-package. Aside from these incompatible location changes, the other changes are additive.

3.3. OSID Providers

OSID Providers implementing these functions will need to relocate them to the new interfaces. OSID Providers are not required to support either Id issuing or creation (and it is not known anyone has at this time).

4. Interoperability Considerations

This change broadens the utility of the Id OSID and strengthens the alignment with other OSIDs. The Id OSID can be used for more detailed management of Ids as well as a tool for assignment of unique codes and sequences.

5. Proposed Interfaces

5.1. osid.id.IdIssueSession

Interface	osid.id.IdIssueSession	
Implements	osid.OsidSession	
Description	This is a simple session used to create new Ids.	
Method	**canIssueIds**	
Description	Tests if this user can issue Ids. A return of true does not guarantee successful authorization. A return of false indicates that it is known create 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 create operations.	
Return	boolean	false if create methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	**issueId**	
Description	Issues a new Id. This method creates a new Id for a predetermined authority and namespace.	
Return	osid.id.Id	the new Id
Errors	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	an authorization failure occurred
Compliance	mandatory	This method must be implemented.

5.2. osid.id.IdAdminSession

Interface	osid.id.IdAdminSession	
Implements	osid.OsidSession	
Description	This session is used to manually create new Ids.	
Method	**issueId**	

Description	Tests if this user can create Ids. A return of true does not guarantee successful authorization. A return of false indicates that it is known create 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 create operations.	
Return	boolean	false if create methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	**getIdFormForCreate**	
Description	Gets the Id form for creating new Ids. A new form should be requested for each create transaction.	
Return	osid.id.IdForm	an IdForm
Errors	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	an authorization failure occurred
Compliance	mandatory	This method must be implemented.
Method	**createId**	
Description	Creates a new Id. a new IdForm should be requested for each create transaction.	
Parameters	osid.id.IdForm	idForm	an IdForm
Return	osid.id.Id	the new Id
Errors	ALREADY_EXISTS	an Id for the authority, namespace, and identifier already exists
	ILLEGAL_STATE	idForm already used in a create transaction
	INVALID_ARGUMENT	the new Id
	NULL_ARGUMENT	idForm is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	an authorization failure occurred
	UNSUPPORTED	the new Id
Compliance	mandatory	This method must be implemented.
Method	**canAliasIds**	
Description	Tests if this user can alias Ids. A return of true does not guarantee successful authorization. A return of false indicates that it is known add 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 alias operations.	
Return	boolean	false if alias methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	**aliasId**	
Description	Makes two Ids equivalent. The primary and equivalent Ids are already known to this service . If the external Id is already mapped to another Id, it is changed to map to the given primary Id. Calls to IdLookupSession.getId(equivalentId) return the primaryId.	
Parameters	osid.id.Id	primaryId	the primary Id
	osid.id.Id	equivalentId	the Id to be made equivalent
Errors	NOT_FOUND	primaryId or equivalentId is not found
	NULL_ARGUMENT	primaryId or equivalentId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	an authorization failure occurred
Compliance	mandatory	This method must be implemented.
Method	**removeAlisas**	
Description	Removes equivalence from two Ids.	
Parameters	osid.id.Id	primaryId	the primary Id
	osid.id.Id	equivalentId	the equivalent Id

Errors	NOT_FOUND	primaryId or equivalentId is not found or equivalentId not mapped to primaryId
	NULL_ARGUMENT	primaryId or equivalentId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	an authorization failure occurred
Compliance	mandatory	This method must be implemented.

5.3. osid.id.IdForm

Interface	osid.id.IdForm	
Implements	osid.OsidSession	
Description	This form provides a means of creating an Id.	
Method	**getAuthorityMetadata**	
Description	Gets the metadata for the authority.	
Return	osid.Metadata	the metadata for the authority
Compliance	mandatory	This method must be implemented.
Method	**setAuthority**	
Description	Sets the authority.	
Parameters	string	authority	the authority
Errors	INVALID_ARGUMENT	authority is invalid
	NO_ACCESS	authority cannot be modified
	NULL_ARGUMENT	authority is null
Compliance	mandatory	This method must be implemented.
Method	**clearAuthority**	
Description	Clears the authority.	
Errors	NO_ACCESS	Metadata.isRequired() is true or Metadata.isReadOnly() is true
Compliance	mandatory	This method must be implemented.
Method	**getIdentifierNamespaceMetadata**	
Description	Gets the metadata for the identifier namespace.	
Return	osid.Metadata	the metadata for the identifier namespace
Compliance	mandatory	This method must be implemented.
Method	**setIdentifierNamespace**	
Description	Sets the identifier namespace.	
Parameters	string	namespace	the identifier namespace
Errors	INVALID_ARGUMENT	namespace is invalid
	NO_ACCESS	namespace cannot be modified
	NULL_ARGUMENT	namespace is null
Compliance	mandatory	This method must be implemented.
Method	**clearIdentiferNamespace**	
Description	Clears the identifier namespace.	
Errors	NO_ACCESS	Metadata.isRequired() is true or Metadata.isReadOnly() is true
Compliance	mandatory	This method must be implemented.
Method	**getIdentifierPrefixMetadata**	
Description	Gets the metadata for the identifier prefix.	
Return	osid.Metadata	the metadata for the identifier prefix
Compliance	mandatory	This method must be implemented.

Method	setIdentifierPrefix		
Description	Sets the identifier prefix.		
Parameters	string	prefix	the identifier prefix
Errors	INVALID_ARGUMENT		prefix is invalid
	NO_ACCESS		prefix cannot be modified
	NULL_ARGUMENT		prefix is null
Compliance	mandatory		This method must be implemented.
Method	clearIdentiferPrefix		
Description	Clears the identifier prefix.		
Errors	NO_ACCESS		Metadata.isRequired() is true or Metadata.isReadOnly() is true
Compliance	mandatory		This method must be implemented.
Method	getIdentifierSuffixMetadata		
Description	Gets the metadata for the identifier prefix.		
Return	osid.Metadata		the metadata for the identifier prefix
Compliance	mandatory		This method must be implemented.
Method	setIdentifierSuffix		
Description	Sets the identifier suffix.		
Parameters	string	suffix	the identifier suffix
Errors	INVALID_ARGUMENT		suffix is invalid
	NO_ACCESS		suffix cannot be modified
	NULL_ARGUMENT		suffix is null
Compliance	mandatory		This method must be implemented.
Method	clearIdentiferSuffix		
Description	Clears the identifier suffix.		
Errors	NO_ACCESS		Metadata.isRequired() is true or Metadata.isReadOnly() is true
Compliance	mandatory		This method must be implemented.
Method	getIdentifierMetadata		
Description	Gets the metadata for the identifier.		
Return	osid.Metadata		the metadata for the identifier
Compliance	mandatory		This method must be implemented.
Method	setIdentifier		
Description	Sets the identifier.		
Parameters	string	identifier	the identifier
Errors	INVALID_ARGUMENT		identifier is invalid
	NO_ACCESS		identifier cannot be modified
	NULL_ARGUMENT		identifier is null
Compliance	mandatory		This method must be implemented.
Method	clearIdentifer		
Description	Clears the identifier.		
Errors	NO_ACCESS		Metadata.isRequired() is true or Metadata.isReadOnly() is true
Compliance	mandatory		This method must be implemented.

6. Example

<div align="center">Simple Id Issue</div>

```
osid.id.Id newId = idIssueSession.issueId();
```

Complex Id Creation

```
osid.id.IdForm form = idAdminSession.getIdFormForCreate();
form.setAuthority("osid.org");
form.setIdentifierNamespace("osr");
form.setIdentifierPrefix("OSR-");
osid.id.Id newId = idAdminSession.createId(form);
```

7. Copyright Statement