OSID V3 Specifications
cataloging package

Version Draft 3

This specifications represent a draft for OSID V3 interface definitions. These definitions may change at any time.

Last Modified: 27 October 2008

prepared by:
Tom Coppeto
OnTapSolutions

	OSID License
Copyright	Copyright © 2008 Massachusetts Institute of Technology. All Rights Reserved.
License	This Work is being provided by the copyright holder(s) subject to the following license. By obtaining, using and/or copying this Work, you agree that you have read, understand, and will comply with the following terms and conditions. This Work and the information contained herein is provided on an ""AS IS"" basis. The Massachusetts Institute of Technology, the Open Knowledge Initiative, and THE AUTHORS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE WORK OR THE USE OR OTHER DEALINGS IN THE WORK. Permission to use, copy and distribute unmodified versions of this Work, for any purpose, without fee or royalty is hereby granted, provided that you include the above copyright notice and the terms of this license on ALL copies of the Work or portions thereof. You may nodify or create Derivatives of this Work only for your internal purposes. You shall not distribute or transfer any such Derivative of this Work to any location or to any third party. For the purposes of this license, Derivative shall mean any derivative of the Work as defined in the United States Copyright Act of 1976, such as a translation or modification. The export of software employing encryption technology may require a specific license from the United States Government. It is the responsibility of any person or organization comtemplating export to obtain such a license before exporting this Work.

Package Description	osid.cataloging package

Interfaces	osid.cataloging.CatalogingProfile
	osid.cataloging.CatalogingManager
	osid.cataloging.CatalogingProfileManager
	osid.cataloging.CatalogSession
	osid.cataloging.CatalogAssignmentSession
	osid.cataloging.CatalogAssignmentNotificationSession
	osid.cataloging.CatalogLookupSession
	osid.cataloging.CatalogSearchSession
	osid.cataloging.CatalogAdminSession
	osid.cataloging.CatalogNotificationSession
	osid.cataloging.CatalogHierarchySession
	osid.cataloging.CatalogHierarchyDesignSession
	osid.cataloging.Catalog
	osid.cataloging.CatalogRecord
	osid.cataloging.CatalogQuery
	osid.cataloging.CatalogQueryRecord
	osid.cataloging.CatalogForm
	osid.cataloging.CatalogFormRecord
	osid.cataloging.CatalogSearch
	osid.cataloging.CatalogSearchRecord
	osid.cataloging.CatalogSearchResults
	osid.cataloging.CatalogSearchResultsRecord
	osid.cataloging.CatalogReceiver
	osid.cataloging.CatalogList
	osid.cataloging.CatalogEntryReceiver

Package	osid.cataloging
Title	Cataloging Open Service Interface Definitions
Version	3.0.0
Description	The Cataloging OSID provides a means for organizing objects into catalogs. An object is represented by an OSID Id. Catalogs can be used for OsidObjects using their Ids or consumer objects whose identifiers are exprsssed via an Id.

Catalogs can be organized into hierarchies for federation. A catalog that is a parent of another catalog includes the Ids of its children.

The Cataloging OSID provides the core definition for the catalogging functions used throughout the OSIDs and aids in the adaptability of cataloging structures. |

Interface	osid.cataloging.CatalogingProfile
Implements	osid.OsidProfile
Description	The cataloging profile describes the interoperability among cataloging services.

Method	supportsCatalog	
Description	Tests for the availability of a cataloging service retrieving Id to Catalog mappings.	
Return	boolean	true if cataloging is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogAssignment	
Description	Tests for the availability of a cataloging service for mapping Ids to Catalogs.	
Return	boolean	true if catalog assignment is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogAssignmentNotification	
Description	Tests for the availability of a cataloging notification service for mapping Ids to Catalogs.	
Return	boolean	true if catalog assignment notification is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogLookup	
Description	Tests for the availability of a catalog lookup service.	
Return	boolean	true if catalog lookup is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogSearch	
Description	Tests for the availability of a catalog search service that defines more comprehensive queries.	
Return	boolean	true if catalog search is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogAdmin	
Description	Tests for the availability of a catalog administration service for the addition and deletion of catalogs.	
Return	boolean	true if catalog administration is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogNotification	
Description	Tests for the availability of a catalog notification service.	
Return	boolean	true if catalog notification is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogHierarchy	
Description	Tests for the availability of a catalog hierarchy traversal service.	
Return	boolean	true if catalog hierarchy traversal is available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	supportsCatalogHierarchyDesign	
Description	Tests for the availability of a catalog hierarchy design service.	
Return	boolean	true if catalog hierarchy design is available, false otherwise
Compliance	mandatory	This method must be implemented in all providers.

Method	supportsConfigurationHierarchySequencing	
Description	Tests if the configuration hierarchy supports node sequencing.	
Return	boolean	true if configuration hierarchy node sequencing is supported, false otherwise
Compliance	mandatory	This method must be implemented.

Method	getCatalogRecordTypes	
Description	Gets the supported Catalog record types.	
Return	osid.type.TypeList	a list containing the supported Catalog record types
Compliance	mandatory	This method must be implemented.
Method	supportsCatalogRecordType	
Description	Tests if the given Catalog record type is supported.	
Parameters	osid.type.Type catalogRecordType	a Type indicating a Catalog record type
Return	boolean	true if the given Type is supported, false otherwise
Compliance	mandatory	This method must be implemented.
Method	getCatalogSearchRecordTypes	
Description	Gets the supported catalog search reciord types.	
Return	osid.type.TypeList	a list containing the supported search record types
Compliance	mandatory	This method must be implemented.
Method	supportsCatalogSearchRecordType	
Description	Tests if the given catalog search record type is supported.	
Parameters	osid.type.Type catalogSearchRecordType	a Type indicating a catalog search record type
Return	boolean	true if the given Type is supported, false otherwise
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogingManager
Implements	osid.OsidManager osid.cataloging.CatalogingProfile
Description	The cataloging manager provides access to cataloging sessions and provides interoperability tests for various aspects of this service. The sessions included in this manager are: • CatalogSession: a session to lookup mappings to catalogs • CatalogAssignmentSession: a session to manage Id to Catalog mappings • CatalogAssignmentNotificationSession: a session to receive notification of changed mappings • CatalogLookupSession: a session to retrieve catalog objects • CatalogSearchSession: a session to search for catalogs • CatalogAdminSession: a session to create, update and delete catalogs • CatalogNotificationSession: a session to receive notifications for changes in catalogs • CatalogHierarchyTraversalSession: a session to traverse hierarchies of catalogs • CatalogHierarchyDesignSession: a session to manage hierarchues of catalogs The cataloging manager also provides a profile for determing the supported search types supported by this service.

Method	getCatalogSession	
Description	Gets the cataloging session for retrieving mappings to catalogs.	
Return	osid.cataloging.CatalogSession	a CatalogSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalog() is false
Compliance	optional	This method must be implemented if supportsCatalog() is true.

Method	getCatalogAssignmentSession	
Description	Gets the cataloging session for adding and removing mappings to catalogs.	
Return	osid.cataloging.CatalogAssignmentSession	a CatalogAssignmentSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogAssignment() is false
Compliance	optional	This method must be implemented if supportsCatalogAssignment() is true.

Method	getCatalogAssignmentNotificationSession	
Description	Gets the notification session for subscribing to changes to catalogs.	
Parameters	osid.cataloging.CatalogEntryReceiver receiver	the notification callback
Return	osid.cataloging.CatalogNotificationSession	a CatalogAssignmentNotificationSession
Errors	NULL_ARGUMENT	receiver is null
	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogAssignmentNotification() is false
Compliance	optional	This method must be implemented if supportsCatalogAssignmentNotification() is true.

Method	getCatalogLookupSession	
Description	Gets the catalog lookup session.	
Return	osid.cataloging.CatalogLookupSession	a CatalogLookupSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogLookup() is false
Compliance	optional	This method must be implemented if supportsCatalogLookup() is true.

Method	getCatalogSearchSession	
Description	Gets the catalog search session.	
Return	osid.cataloging.CatalogSearchSession	a CatalogSearchSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogSearch() is false
Compliance	optional	This method must be implemented if supportsCatalogSearch() is true.

Method	getCatalogAdminSession	
Description	Gets the catalog administrative session for creating, updating and deleting catalogs.	
Return	osid.cataloging.CatalogAdminSession	a CatalogAdminSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogAdmin() is false
Compliance	optional	This method must be implemented if supportsCatalogAdmin() is true.

Method	getCatalogNotificationSession	
Description	Gets the notification session for subscribing to changes to catalogs.	
Parameters	osid.cataloging.CatalogReceiver receiver	the notification callback
Return	osid.cataloging.CatalogNotificationSession	a CatalogNotificationSession
Errors	NULL_ARGUMENT	receiver is null
	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogNotification() is false
Compliance	optional	This method must be implemented if supportsCatalogNotification() is true.

Method	getCatalogHierarchySession	
Description	Gets the catalog hierarchy traversal session.	
Return	osid.cataloging.CatalogHierarchySession	a CatalogHierarchySession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogHierarchy() is false
Compliance	optional	This method must be implemented if supportsCatalogHierarchy() is true.

Method	getCatalogHierarchyDesignSession	
Description	Gets the catalog hierarchy design session.	
Return	osid.cataloging.CatalogHierarchyDesignSession	a CatalogHierarchyDesignSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsCatalogHierarchyDesign() is false
Compliance	optional	This method must be implemented if supportsCatalogHierarchyDesign() is true.

Interface	osid.cataloging.CatalogingProxyManager
Implements	osid.OsidProxyManager osid.cataloging.CatalogingProfile
Description	The cataloging manager provides access to cataloging sessions and provides interoperability tests for various aspects of this service. Methods in this manager support the passing of an Authentication object for the purposes of proxy authentication. • CatalogSession: a session to lookup mappings to catalogs • CatalogAssignmentSession: a session to manage Id to Catalog mappings • CatalogAssignmentNotificationSession: a session to receive notification of changed mappings • CatalogLookupSession: a session to retrieve catalog objects • CatalogSearchSession: a session to search for catalogs • CatalogAdminSession: a session to create, update and delete catalogs • CatalogNotificationSession: a session to receive notifications for changes in catalogs • CatalogHierarchyTraversalSession: a session to traverse hierarchies of catalogs • CatalogHierarchyDesignSession: a session to manage hierarchues of catalogs The cataloging manager also provides a profile for determing the supported search types supported by this service.

Method	getCatalogSession	
Description	Gets the catalog session for retrieving Id to Catalog mappings.	
Parameters	osid.authentication.Authentication authentication	proxy authentication
Return	osid.cataloging.CatalogSession	a CatalogSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsCatalog() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsCatalog() is true.

Method	getCatalogAssignmentSession	
Description	Gets the catalog session for mapping Ids to Catalogs.	
Parameters	osid.authentication.Authentication authentication	proxy authentication
Return	osid.cataloging.CatalogAssignmentSession	a CatalogAssignmentSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsCatalogAssignment() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsCatalogAssignment() is true.

Method	getCatalogAssignmentNotificationSession	
Description	Gets the catalog session for mapping Ids to Catalogs.	
Parameters	osid.cataloging.CatalogEntryReceiver receiver	the notification callback
	osid.authentication.Authentication authentication	proxy authentication
Return	osid.cataloging.CatalogAssignmentNotificationSession	a CatalogAssignmentNotificationSession
Errors	NULL_ARGUMENT	receiver or authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsCatalogAssignmentNotification() is
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsCatalogAssignmentNotification() is true.

Method	getCatalogLookupSession	
Description	Gets the catalog lookup session.	
Parameters	osid.authentication.Authentication authentication	proxy authentication
Return	osid.cataloging.CatalogLookupSession	a CatalogLookupSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsCatalogLookup() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsCatalogLookup() is true.

Method	getCatalogSearchSession	
Description	Gets the catalog search session.	
Parameters	osid.authentication.Authentication authentication	proxy authentication
Return	osid.cataloging.CatalogSearchSession	a CatalogSearchSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsCatalogSearch() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsCatalogSearch() is true.

Method	getCatalogAdminSession	
Description	Gets the catalog administrative session for creating, updating and deleting catalogs.	
Parameters	osid.authentication.Authentication authentication	proxy authentication
Return	osid.cataloging.CatalogAdminSession	a CatalogAdminSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsCatalogAdmin() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsCatalogAdmin() is true.

Method	getCatalogNotificationSession		
Description	Gets the notification session for subscribing to changes to catalogs.		
Parameters	osid.cataloging.CatalogReceiver	receiver	the notification callback
	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.cataloging.CatalogNotificationSession		a CatalogNotificationSession
Errors	NULL_ARGUMENT		receiver or authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsCatalogNotification() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsCatalogNotification() is true.
Method	getCatalogHierarchySession		
Description	Gets the catalog hierarchy traversal session.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.cataloging.CatalogHierarchySession		a CatalogHierarchySession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsCatalogHierarchy() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsCatalogHierarchy() is true.
Method	getCatalogHierarchyDesignSession		
Description	Gets the catalog hierarchy design session.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.cataloging.CatalogHierarchyDesignSession		a CatalogHierarchyDesignSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsCatalogHierarchyDesign() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsCatalogHierarchyDesign() is true.

Interface	osid.cataloging.CatalogSession
Implements	osid.OsidSession
Description	This session provides methods to retrieve OSID Id to Catalog mappings. An Id may appear in multiple Catalogs. Each Catalog may have its own authorizations as to who is allowed to look at it. This lookup session defines several views: • federated view: entries are accessible from the specified Catalog and any descendant dictionaries in the Catalog hierarchy • isolated view: entries are accessible from the specified Catalog only • comparative view: elements may be silently omitted or re-ordered • plenary view: provides a complete result set or is an error condition Generally, the comparative view should be used for most applications as it permits operation even if there a particular element is inaccessible. For example, a hierarchy output can be plugged into a lookup method to retrieve all objects known to a hierarchy, but it may not be necessary to break execution if a node from the hierarchy no longer exists. However, some administrative applications may need to know whether it had retrieved an entire set of objects and may sacrifice some interoperability for the sake of

Method	useComparativeCatalogView	
Description	The returns from the lookup methods may omit or translate elements based on this session, such as authorization, and not result in an error. This view is used when greater interoperability is desired at the expense of precision.	
Compliance	mandatory	This method is must be implemented.

Method	usePlenaryCatalogView	
Description	A complete view of the Id and Catalog returns is desired. Methods will return what is requested or result in an error. This view is used when greater precision is desired at the expense of interoperability.	
Compliance	mandatory	This method is must be implemented.

Method	useFederatedCatalogView	
Description	Federates the view for methods in this session. A federated view will include entries from descendant catalogs in the catalog hierarchy.	
Compliance	mandatory	This method is must be implemented.

Method	useIsolatedCatalogView	
Description	Isolates the view for methods in this session. An isolated view restricts lookups to the specified catalog only.	
Compliance	mandatory	This method is must be implemented.

Method	canLookupMappings	
Description	Tests if this user can perform lookups of Id to Catalog mappings. A return of true does not guarantee successful authorization. A return of false indicates that it is known lookup 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	boolean	false if looking up mappings is not authorized, true otherwise
Compliance	mandatory	This method must be implemented.

Method	getIdsByCatalog	
Description	Gets the list of Ids map to a Catalog.	
Parameters	osid.id.Id	catalogId	a catalog Id
Return	osid.id.IdList	list of Ids mapped the given catalogId
Errors	NOT_FOUND	catalogId is not found
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	getIdsByCatalogs		
Description	Gets the list of Ids map to a lst of Catalogs.		
Parameters	osid.id.IdList	catalogIdList	an Id
Return	osid.id.IdList		list of catalogs containing the given Id
Errors	NULL_ARGUMENT		idList is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Method	getCatalogIdsById		
Description	Gets the Catalog Ids mapped to an Id.		
Parameters	osid.id.Id	id	an Id
Return	osid.id.IdList		list of catalogs Ids containing the given Id
Errors	NULL_ARGUMENT		id is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Method	getCatalogsById		
Description	Gets the Catalogs mapped to an Id.		
Parameters	osid.id.Id	id	an Id
Return	osid.cataloging.CatalogList		list of catalogs containing the given Id
Errors	NULL_ARGUMENT		id is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogAssignmentSession	
Implements	osid.OsidSession	
Description	This session provides methods to assign OSID Ids to Catalogs. An Id may appear in multiple Catalogs and removing the last reference to an Id is the equivalent of deleting it. Each Catalog may have its own authorizations as to who is allowed to operate on it.	
Method	canAssignCatalogs	
Description	Tests if this user can perform alter Id/Catalog 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	boolean	false if mapping is not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	assignId	
Description	Adds an Id to a Catalog.	
Parameters	osid.id.Id	id	an Id
	osid.id.Id	catalogId	the Id of the Catalog
Errors	NOT_FOUND	id or catalogId not found
	NULL_ARGUMENT	id or catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.
Method	reassignId	
Description	Moves an Id to a Catalog.	
Parameters	osid.id.Id	id	an Id
	osid.id.Id	catalogId	the Id of the Catalog
Errors	NOT_FOUND	id or catalogId not found
	NULL_ARGUMENT	id or catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.
Method	unassignId	
Description	Removes an Id from a Catalog.	
Parameters	osid.id.Id	id	an Id
	osid.id.Id	catalogId	the Id of the Catalog
Errors	NOT_FOUND	id or catalogId not found
	NULL_ARGUMENT	id or catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogAssignmentNotificationSession	
Implements	osid.OsidSession	
Description	This session defines methods to receive notifications on adds/changes to a catalog assignment. This session is intended for consumers needing to synchronize their state with this service without the use of polling. Notifications are cancelled when this session is closed. Notifications are triggered with changes to assignments of Ids to this catalog. For notifications of changes to the Catalog object use CatalogNotificationSession.	
Method	canRegisterForCatalogEntryNotifications	
Description	Tests if this user can register for Catalog entry notifications. A return of true does not guarantee successful authorization. A return of false indicates that it is known all 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 notification operations.	
Return	boolean	false if notification methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	useFederatedCatalogView	
Description	Federates the view for methods in this session. A federated view will include entries from parent catalogs in the catalog hierarchy.	
Compliance	mandatory	This method is must be implemented.
Method	useIsolatedCatalogView	
Description	Isolates the view for methods in this session. An isolated view restricts notifications for entries to this catalog only.	
Compliance	mandatory	This method is must be implemented.
Method	registerForNewCatalogEntries	
Description	Register for notifications of new catalogs. CatalogEntryReceiver.newCatalogEntry() is invoked when a new Catalog entry is created.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog node to monitor
Errors	NOT_FOUND	a catalog node was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.
Method	registerForDeletedCatalogEntries	
Description	Registers for notification of deleted catalogs entries. CatalogEntryReceiver.deletedCatalogEntry() is invoked when a catalog entry is deleted.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog node to monitor
Errors	NOT_FOUND	a catalog node was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogLookupSession
Implements	osid.OsidSession
Description	This session provides methods for retrieving Catalog objects. The Catalog represents a collection of OSID Ids.

This session defines views that offer differing behaviors when retrieving multiple objects.

• comparative view: elements may be silently omitted or re-ordered
• plenary view: provides a complete set or is an error condition

Generally, the comparative view should be used for most applications as it permits operation even if there is data that cannot be accessed.. For example, a browsing application may only need to examine the Catalogs it can access, without breaking execution. However, an assessment may only be useful if all Catalogs referenced by it are available, and a test-taking applicationmay sacrifice some interoperability for the sake of precision.

Method	canLookupCatalogs	
Description	Tests if this user can perform Catalog lookups. A return of true does not guarantee successful authorization. A return of false indicates that it is known all 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	boolean	false if lookup methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	useComparativeCatalogView	
Description	The returns from the lookup methods may omit or translate elements based on this session, such as authorization, and not result in an error. This view is used when greater interoperability is desired at the expense of precision.	
Compliance	mandatory	This method is must be implemented.
Method	usePlenaryCatalogView	
Description	A complete view of the Catalog returns is desired. Methods will return what is requested or result in an error. This view is used when greater precision is desired at the expense of interoperability.	
Compliance	mandatory	This method is must be implemented.
Method	getCatalog	
Description	Gets the Catalog specified by its Id. In plenary mode, the exact Id is found or a NOT_FOUND results. Otherwise, the returned Catalog may have a different Id than requested, such as the case where a duplicate Id was assigned to a Catalog and retained for compatibility.	
Parameters	osid.id.Id	catalogId	Id of the Catalog
Return	osid.cataloging.Catalog	the catalog
Errors	NOT_FOUND	catalogId not found
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method is must be implemented.

Method	getCatalogsByIds	
Description	Gets a CatalogList corresponding to the given IdList. In plenary mode, the returned list contains all of the catalogs specified in the Id list, in the order of the list, including duplicates, or an error results if an Id in the supplied list is not found or inaccessible. Otherwise, inaccessible Catalogs may be omitted from the list and may present the elements in any order including returning a unique set.	
Parameters	osid.id.IdList	catalogIdList	the list of Ids to rerieve
Return	osid.cataloging.CatalogList		the returned Catalog list
Errors	NOT_FOUND	an Id was not found
	NULL_ARGUMENT	catalogIdList is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	getCatalogsByGenusType	
Description	Gets a CatalogList corresponding to the given catalog genus Type which does not include catalogs of genus types derived from the specified Type. In plenary mode, the returned list contains all known catalogs or an error results. Otherwise, the returned list may contain only those catalogs that are accessible through this session. In both cases, the order of the set is not specified.	
Parameters	osid.type.Type	catalogGenusType	a catalog genus type
Return	osid.cataloging.CatalogList		the returned Catalog list
Errors	NULL_ARGUMENT	catalogGenusType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	getCatalogsByParentGenusType	
Description	Gets a CatalogList corresponding to the given catalog genus Type and include any additional catalogs with genus types derived from the specified Type. In plenary mode, the returned list contains all known catalogs or an error results. Otherwise, the returned list may contain only those catalogs that are accessible through this session. In both cases, the order of the set is not specified.	
Parameters	osid.type.Type	catalogGenusType	a catalog genus type
Return	osid.cataloging.CatalogList		the returned Catalog list
Errors	NULL_ARGUMENT	catalogGenusType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	getCatalogsByRecordType	
Description	Gets a CatalogList containing the given subject record Type. In plenary mode, the returned list contains all known subjects or an error results. Otherwise, the returned list may contain only those catalogs that are accessible through this session. In both cases, the order of the set is not specified.	
Parameters	osid.type.Type	catalogRecordType	a catalog interface type
Return	osid.cataloging.CatalogList		the returned Catalog list
Errors	NULL_ARGUMENT	catalogInterfaceType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	getCatalogs	
Description	Gets all Catalogs. In plenary mode, the returned list contains all known catalogs or an error results. Otherwise, the returned list may contain only those catalogs that are accessible through this session. In both cases, the order of the set is not specified.	
Return	osid.cataloging.CatalogList	a list of Catalogs
Errors	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogSearchSession	
Implements	osid.OsidSession	
Description	This session provides methods for searching Catalog objects. The search query is constructed using the CatalogQuery interface.The catalog interface Type also specifies the interface for the catalog query. getCatalogsByQuery() is the basic search method and returns a list of Catalog elements. A more advanced search may be performed with getCatalogsBySearch(). It accepts a CatalogSearch interface in addition to the query interface for the purpose of specifying additional options affecting the entire search, such as ordering. getCatalogsBySearch() returns a CatalogSearchResults interface that can be used to access the resulting CatalogList or be used to perform a search within the result set through CatalogSearch. Catalogs may have a record query interface indicated by their respective record interface types. The record query interface is accessed via the CatalogQuery.	
Method	**canSearchCatalogs**	
Description	Tests if this user can perform Catalog searches. A return of true does not guarantee successful authorization. A return of false indicates that it is known all 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 search operations to unauthorized users.	
Return	boolean	false if search methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	**getCatalogQuery**	
Description	Gets a catalog query interface. The returned query will not have an extension query.	
Return	osid.cataloging.CatalogQuery	the catalog query
Compliance	mandatory	This method must be implemented.
Method	**getCatalogsByQuery**	
Description	Gets a list of Catalogs matching the given search interface.	
Parameters	osid.cataloging.CatalogQuery catalogQuery	the search query
Return	osid.cataloging.CatalogList	the returned CatalogList
Errors	NULL_ARGUMENT	catalogQuery is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	catalogQuery is not of this service
Compliance	mandatory	This method must be implemented.
Method	**getCatalogSearch**	
Description	Gets a catalog search interface. The returned query only makes available the core CatalogSearch interface and does not support additional interface types. getCatalogSearchForInterfaceType() should be used if additional interface types are required.	
Return	osid.cataloging.CatalogSearch	the catalog search interface
Compliance	mandatory	This method must be implemented.

Method	getCatalogsSearchOrder	
Description	Gets a subject search order interface. The CatalogSearchOrder is supplied to a CatalogSearch to specify the ordering of results.	
Return	osid.cataloging.CatalogSearchOrder	the catalog search order interface
Compliance	mandatory	This method must be implemented.

Method	getCatalogsBySearch	
Description	Gets the search results matching the given search interface.	
Parameters	osid.cataloging.CatalogQuery	catalogQuery	the search query
	osid.cataloging.CatalogSearch	catalogSearch	the search interface
Return	osid.cataloging.CatalogSearchResults	the search results
Errors	NULL_ARGUMENT	catalogQuery or catalogSearch is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	catalogQuery or catalogSearch is not of this service
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogAdminSession	
Implements	osid.OsidSession	
Description	This session creates and removes catalogs. The data for create and update is provided by the consumer via the form object.	
Method	**canCreateCatalogs**	
Description	Tests if this user can create Catalogs. A return of true does not guarantee successful authorization. A return of false indicates that it is known creating a Catalog will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer create operations to unauthorized users.	
Return	boolean	false if Catalog creation is not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	**canCreateCatalogWithRecordTypes**	
Description	Tests if this user can create a single Catalog using the desired record interface types. While CatalogingManager.getCatalogRecordTypes() can be used to examine which record interfaces are supported, this method tests which record(s) are required for creating a specific Catalog. Providing an empty array tests if a Catalog can be created with no records.	
Parameters	osid.type.Type[]	catalogRecordTypes	array of types
Return	boolean	true if Catalog creation using the specified record Types is supported, false otherwise
Errors	NULL_ARGUMENT	catalogRecordTypes is null
Compliance	mandatory	This method must be implemented.
Method	**getCatalogFormForCreate**	
Description	Gets the catalog form for creating new catalogs. A new form should be requested for each create transaction.	
Return	osid.cataloging.CatalogForm	the catalog form
Compliance	mandatory	This method must be implemented.
Method	**createCatalog**	
Description	Creates a new Catalog.	
Parameters	osid.cataloging.CatalogForm	catalogForm	the forms for this Catalog
Return	osid.cataloging.Catalog	the new Catalog
Errors	ALREADY_EXISTS	attempt at duplicating a property the underlying system is enforcing to be unique
	INVALID_ARGUMENT	one or more of the form elements is invalid
	NULL_ARGUMENT	catalogForm is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	catalogForm is not of this service
Compliance	mandatory	This method must be implemented.
Method	**canUpdateCatalogs**	
Description	Tests if this user can update Catalogs. A return of true does not guarantee successful authorization. A return of false indicates that it is known updating a Catalog will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer update operations to unauthorized users.	
Return	boolean	false if Catalog modification is not authorized, true otherwise
Compliance	mandatory	This method must be implemented.

Method	canUpdateCatalog		
Description	Tests if this user can update a specified Catalog. A return of true does not guarantee successful authorization. A return of false indicates that it is known updating the Catalog will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer update operations to unauthorized users.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog
Return	boolean		false if catalog modification is not authorized, true otherwise
Errors	NULL_ARGUMENT		catalogId is null
Compliance	mandatory		This method must be implemented.
Provider Notes	If the catalogId is not found, then it is acceptable to return false to indicate the lack of an update available.		

Method	getCatalogFormForUpdate		
Description	Gets the catalog form for updating an existing catalog. A new catalog form should be requested for each update transaction.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog
Return	osid.cataloging.CatalogForm		the catalog form
Errors	NULL_ARGUMENT		catalogId is null
Compliance	mandatory		This method must be implemented.

Method	updateCatalog		
Description	Updates an existing catalog.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog
	osid.cataloging.CatalogForm	catalogForm	the form containing the elements to be updated
Errors	INVALID_ARGUMENT		the form contains an invalid value
	NOT_FOUND		catalogId is not found
	NULL_ARGUMENT		catalogId or catalogForm is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
	UNSUPPORTED		catalogForm is not supported
Compliance	mandatory		This method must be implemented.

Method	canDeleteCatalogs		
Description	Tests if this user can delete Catalogs. A return of true does not guarantee successful authorization. A return of false indicates that it is known deleting a Catalog will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer delete operations to unauthorized users.		
Return	boolean		false if Catalog deletion is not authorized, true otherwise
Compliance	mandatory		This method must be implemented.

Method	canDeleteCatalog		
Description	Tests if this user can delete a specified Catalog. A return of true does not guarantee successful authorization. A return of false indicates that it is known deleting the Catalog will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer delete operations to unauthorized users.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog
Return	boolean		false if deletion of this Catalog is not authorized, true otherwise
Errors	NULL_ARGUMENT		catalogId is null
Compliance	mandatory		This method must be implemented.
Provider Notes	If the catalogId is not found, then it is acceptable to return false to indicate the lack of an delete available.		

Method	deleteCatalog		
Description	Deletes a Catalog.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog to remove
Errors	NOT_FOUND		catalogId not found
	NULL_ARGUMENT		catalogId is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.
Method	addIdToCatalog		
Description	Adds an Id to a Catalog for the purpose of creating compatibility. The primary Id of the Catalog is determined by the provider. The new Id performs as an alias to the primary Id.		
Parameters	osid.id.Id	catalogId	the Id of a Catalog
	osid.id.Id	aliasId	the alias Id
Errors	ALREADY_EXISTS		aliasId is already assigned
	NOT_FOUND		catalogId not found
	NULL_ARGUMENT		catalogId or aliasId is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogNotificationSession	
Implements	osid.OsidSession	
Description	This session defines methods to receive notifications on adds/changes to Catalog objects. This session is intended for consumers needing to synchronize their state with this service without the use of polling. Notifications are cancelled when this session is closed. Notifications are triggered with changes to the Catalog object itself. Adding and removing Ids result in notifications available from the notification session for catalog entries.	
Method	canRegisterForCatalogNotifications	
Description	Tests if this user can register for Catalog notifications. A return of true does not guarantee successful authorization. A return of false indicates that it is known all 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 notification operations.	
Return	boolean	false if notification methods are not authorized, true otherwise
Compliance	mandatory	This method must be implemented.
Method	registerForNewCatalogs	
Description	Register for notifications of new catalogs. CatalogReceiver.newCatalog() is invoked when a new Catalog is created.	
Errors	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.
Method	registerForNewCatalogAncestors	
Description	Registers for notification of an updated hierarchy structure that introduces a new ancestor of the specified catalog. CatalogReceiver.newAncestorCatalog() is invoked when the specified catalog node gets a new ancestor.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog node to monitor
Errors	NOT_FOUND	a catalog node was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.
Method	registerForNewCatalogDescendants	
Description	Registers for notification of an updated hierarchy structure that introduces a new descendant of the specified catalog. CatalogReceiver.newDescendantCatalog() is invoked when the specified catalog node gets a new descendant.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog node to monitor
Errors	NOT_FOUND	a catalog node was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.
Method	registerForChangedCatalogs	
Description	Registers for notification of updated catalogs. CatalogReceiver.changedCatalog() is invoked when a catalog is changed.	
Errors	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	registerForChangedCatalog	
Description	Registers for notification of an updated catalog. CatalogReceiver.changedCatalog() is invoked when the specified catalog is changed.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog to monitor
Errors	NOT_FOUND	a catalog was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	registerForDeletedCatalogs	
Description	Registers for notification of deleted catalogs. CatalogReceiver.deletedCatalog() is invoked when a catalog is deleted.	
Errors	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	registerForDeletedCatalog	
Description	Registers for notification of a deleted catalog. CatalogReceiver.deletedCatalog() is invoked when the specified catalog is deleted.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog to monitor
Errors	NOT_FOUND	a catalog was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	registerForDeletedCatalogAncestors	
Description	Registers for notification of an updated hierarchy structure that removes an ancestor of the specified catalog. CatalogReceiver.deletedAncestor() is invoked when the specified catalog node loses an ancestor.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog node to monitor
Errors	NOT_FOUND	a catalog node was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Method	registerForDeletedCatalogDescendants	
Description	Registers for notification of an updated hierarchy structure that removes a descendant of the specified catalog. CatalogReceiver.deletedDescendant() is invoked when the specified catalog node loses a descendant.	
Parameters	osid.id.Id	catalogId	the Id of the Catalog node to monitor
Errors	NOT_FOUND	a catalog node was not found identified by the given Id
	NULL_ARGUMENT	catalogId is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogHierarchySession
Implements	osid.hierarchy.HierarchyTraversalSession
Description	This session defines methods for traversing a hiercrhy of Catalog objects. Each node in the hierarchy is a unique Catalog. The hierarchy may be traversed recursively to establish the tree structure through getParents() and getChildren(). To relate these Ids to another OSID, getAncestors() and getDescendants() can be used for retrievals that can be used for bulk lookups in other OSIDs. Any Catalog available in the Catalog OSID is known to this hierarchy but does not appear in the hierarchy traversal until added as a root node or a child of another node. A user may not be authorized to traverse the entire hierarchy. Parts of the hierarchy may be made invisible through omission from the returns of getParents() or getChildren() in lieu of a PERMISSION_DENIED error that may disrupt the traversal through authorized pathways. This session defines views that offer differing behaviors when retrieving multiple objects. • comparative view: catalog elements may be silently omitted or re-ordered • plenary view: provides a complete set or is an error condition

Method	useComparativeCatalogView
Description	The returns from the catalog methods may omit or translate elements based on this session, such as authorization, and not result in an error. This view is used when greater interoperability is desired at the expense of precision.
Compliance	mandatory — This method is must be implemented.

Method	usePlenaryCatalogView
Description	A complete view of the Catalog returns is desired. Methods will return what is requested or result in an error. This view is used when greater precision is desired at the expense of interoperability.
Compliance	mandatory — This method is must be implemented.

Method	getRootCatalogs
Description	Gets the root catalogs in the catalog hierarchy. A node with no parents is an orphan. While all catalog Ids are known to the hierarchy, an orphan does not appear in the hierarchy unless explicitly added as a root node or child of another node.
Return	osid.cataloging.CatalogList — the root catalogs
Errors	OPERATION_FAILED — unable to complete request PERMISSION_DENIED — authorization failure
Compliance	mandatory — This method is must be implemented.

Method	getParentCatalogs
Description	Gets the parent catalogs of the given id.
Parameters	osid.id.Id — catalogId — the Id of the Catalog to query
Return	osid.cataloging.CatalogList — the parent catalogs of the id
Errors	NOT_FOUND — a Catalog identified by Id is not found NULL_ARGUMENT — catalogId is null OPERATION_FAILED — unable to complete request PERMISSION_DENIED — authorization failure
Compliance	mandatory — This method must be implemented.

Method	getAncestorCatalogs		
Description	Gets the ancestor catalogs of the given id.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog to query
	cardinal	levels	the maximum number of levels to include. A value of 1 returns the same set as getParentCatalogs().
Return	osid.cataloging.CatalogList		the ancestor catalogs of the id
Errors	NOT_FOUND		a Catalog identified by Id is not found
	NULL_ARGUMENT		catalogId is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Method	getChildCatalogs		
Description	Gets the child catalogs of the given id.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog to query
Return	osid.cataloging.CatalogList		the child catalogs of the id
Errors	NOT_FOUND		a Catalog identified by Id is not found
	NULL_ARGUMENT		catalogId is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Method	getDescendantCatalogs		
Description	Gets the descendant catalogs of the given id.		
Parameters	osid.id.Id	catalogId	the Id of the Catalog to query
	cardinal	levels	the maximum number of levels to include. A value of 1 returns the same set as getChildCatalogs().
Return	osid.cataloging.CatalogList		the decsendant catalogs of the id
Errors	NOT_FOUND		a Catalog identified by Id is not found
	NULL_ARGUMENT		catalogId is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogHierarchyDesignSession
Implements	osid.hierarchy.HierarchyDesignSession
Description	This session manages a hierarchy of catalogs. Catalogs may be organized into a hierarchy for organizing or federating. A parent Catalog includes all of the Ids of its children such that a single root node contains all of the Ids of the federation.

Interface	osid.cataloging.Catalog		
Implements	osid.OsidCatalog		
Description	A Catalog represents a collection of entries. Like all OSID objects, a Catalog is identified by its Id and any persisted references should use the Id. getCatalogRecord() should be used to retrieve any record corresponding to arecord Type. The existence of the interface must not be assumed until requested at which point it is safe to cast into the interface indicated by the type.		
	getCatalogInterface		
Description	Gets the record corresponding to the given Catalog record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces. The catalogRecordType may be the Type returned in getRecordTypes() or any of its parents in a Type hierarchy where hasRecordType(catalogRecordType) is true.		
Parameters	osid.type.Type	catalogRecordType	a type of the record to retrieve
Return	osid.cataloging.CatalogRecord		the catalog record
Errors	NULL_ARGUMENT		catalogRecordType is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure occurred
	UNSUPPORTED		hasInterfaceType(catalogRecordType) is false
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogRecord	
Implements	osid.OsidRecord	
Description	A record for a Catalog. The methods specified by the record type are available through the underlying object.	
	getCatalog	
Description	Gets the Catalog from which this record originated.	
Return	osid.cataloging.Catalog	the catalog
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogQuery		
Implements	osid.OsidCatalogQuery		
Description	This is the query interface for searching catalogs. Each method specifies an AND term while multiple invocations of the same method produces a nested OR.		
Method	matchId		
Description	Matches an Id in this catalog. Multiple Ids are treated as a boolean OR.		
Parameters	osid.id.Id	id	Id to match
	boolean	match	true if for a positive match, false for negative match
Errors	NULL_ARGUMENT		id is null
Compliance	mandatory		This method must be implemented.
Method	getCatalogQueryRecord		
Description	Gets the record query interface corresponding to the given Catalog record Type. Multiple record retrievals produce a boolean OR term.		
Parameters	osid.type.Type	catalogRecordType	a catalog record type
Return	osid.cataloging.CatalogQueryRecord		the catalog query record
Errors	NULL_ARGUMENT		catalogRecordType is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure occurred
	UNSUPPORTED		hasRecordType(catalogRecordType) is false
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogQueryRecord	
Implements	osid.OsidRecord	
Description	A record for a CatalogQuery. The methods specified by the record type are available through the underlying object.	
	getCatalogQuery	
Description	Gets the CatalogQuery from which this record originated.	
Return	osid.cataloging.CatalogQuery	the catalog query
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogForm	
Implements	osid.OsidCatalogForm	
Description	This is the form for creating and updating Catalogs. Like all OsidForm objects, various data elements may be set here for use in the create and update methods in the CatalogAdminSession. For each data element that may be set, metadata may be examined to provide display hints or data constraints.	

Method	getCatalogFormRecord	
Description	Gets the CatalogFormRecord interface corresponding to the given catalog record interface Type.	
Parameters	osid.type.Type	catalogRecordType	a catalog record type
Return	osid.cataloging.CatalogFormRecord	the record
Errors	NULL_ARGUMENT	catalogRecordType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure occurred
	UNSUPPORTED	hasRecordType(catalogRecordType) is false
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogFormRecord	
Implements	osid.OsidRecord	
Description	A record for a CatalogForm. The methods specified by the record type are available through the underlying object.	
getCatalogForm		
Description	Gets the CatalogForm from which this record originated.	
Return	osid.cataloging.CatalogForm	the catalog form
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogSearchOrder	
Implements	osid.OsidSearchOrder	
Description	An interface for specifying the ordering of search results.	
Method	getCatalogSearchOrderRecord	
Description	Gets the catalog search order record corresponding to the given subject record Type. Multiple retrievals return the same underlying object.	
Parameters	osid.type.Type	catalogRecordType	a catalog record type
Return	osid.cataloging.CatalogSearchOrderRecord	the catalog search order record interface
Errors	NULL_ARGUMENT	catalogRecordType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure occurred
	UNSUPPORTED	hasRecordType(catalogRecordType) is false
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogSearchOrderRecord
Implements	osid.OsidRecord
Description	A record for a CatalogSearchOrder. The methods specified by the record type are available through the underlying object.

getCatalogSearchOrder		
Description	Gets the CatalogSearchOrder from which this record originated.	
Return	osid.cataloging.CatalogSearchOrder	the catalog search order
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogSearch		
Implements	osid.OsidSearch		
Description	The search interface for governing the search query for Catalogs.		
Method	**searchWithinCatalogResults**		
Description	Execute this search using a previous search result.		
Parameters	osid.cataloging.CatalogSearchResults	results	results from a query
Errors	INVALID_ARGUMENT		results is not valid
	NULL_ARGUMENT		results is null
Compliance	mandatory		This method must be implemented.
Method	**searchAmongCatalogs**		
Description	Execute this search among the given list of catalogs.		
Parameters	osid.id.Id[]	catalogIds	list of catalogs
Errors	NULL_ARGUMENT		catalogIds is null
Compliance	mandatory		This method must be implemented.
Method	**orderCatalogResults**		
Description	Specify an ordering to the search results.		
Parameters	osid.cataloging.CatalogSearchOrder	catalogSearchOrder	catalog search order
Errors	NULL_ARGUMENT		catalogSearchOrder is null
	UNSUPPORTED		catalogSearchOrder is not of this service
Compliance	mandatory		This method must be implemented.
Method	**getCatalogSearchRecord**		
Description	Gets the record corresponding to the given catalog search record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces.		
Parameters	osid.type.Type	catalogSearchRecordType	a catalog search record type
Return	osid.cataloging.CatalogSearchRecord		the catalog search interface
Errors	NULL_ARGUMENT		catalogSearchRecordType is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authorization failure occurred
	UNSUPPORTED		hasSearchRecordType(catalogSearchRecordType) is false
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogSearchRecord	
Implements	osid.OsidSearchRecord	
Description	A record for a CatalogSearch. The methods specified by the record type are available through the underlying object.	
	getCatalogSearch	
Description	Gets the CatalogSearch from which this record originated.	
Return	osid.cataloging.CatalogSearch	the catalog search
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogSearchResults	
Implements	osid.OsidSearchResults	
Description	This interface provides a means to capture results of a search and is used as a vehicle to perform a search within a previous result set.	
Method	getCatalogs	
Description	Gets the catalog list resulting from the search.	
Return	osid.cataloging.CatalogList	the catalogs list
Compliance	mandatory	This method must be implemented.
Method	getCatalogSearchResultsRecord	
Description	Gets the record corresponding to the given catalog search record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces.	
Parameters	osid.type.Type	catalogSearchRecordType	a catalog search record type
Return	osid.cataloging.CatalogSearchResultsRecord	the catalog search interface
Errors	NULL_ARGUMENT	catalogSearchRecordType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure occurred
	UNSUPPORTED	hasSearchRecordType(catalogSearchRecordType) is false
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogSearchResultsRecord	
Implements	osid.OsidSearchRecord	
Description	A record for a CatalogSearchResults. The methods specified by the record type are available through the underlying object.	
	getCatalogSearchResults	
Description	Gets the CatalogSearchResults from which this record originated.	
Return	osid.cataloging.CatalogSearchResults	the catalog search results
Compliance	mandatory	This method must be implemented.

Interface	osid.cataloging.CatalogReceiver		
Implements	osid.OsidReceiver		
Description	The catalog receiver is the consumer supplied interface for receiving notifications pertaining to new, updated or deleted Catalog objects.		
Method	**newCatalog**		
Description	The callback for notifications of new catalogs.		
Parameters	osid.id.Id	catalogId	the Id of the new Catalog
Compliance	mandatory		This method must be implemented.
Method	**newAncestorCatalog**		
Description	The callback for notifications of new ancestors of a catalog.		
Parameters	osid.id.Id	catalogId	the Id of the registered Catalog
	osid.id.Id	ancestorId	the Id of the new ancestor catalog
Compliance	mandatory		This method must be implemented.
Method	**newDescendantCatalog**		
Description	The callback for notifications of new descendant of a catalog.		
Parameters	osid.id.Id	catalogId	the Id of the registered Catalog
	osid.id.Id	descendantId	the Id of the new descendant catalog
Compliance	mandatory		This method must be implemented.
Method	**changedCatalog**		
Description	The callback for notification of updated catalogs.		
Parameters	osid.id.Id	catalogId	the Id of the updated Catalog
Compliance	mandatory		This method must be implemented.
Method	**deletedCatalog**		
Description	the callback for notification of deleted catalogs.		
Parameters	osid.id.Id	catalogId	the Id of the deleted Catalog
Compliance	mandatory		This method must be implemented.
Method	**deletedAncestorCatalog**		
Description	The callback for notifications of deleted ancestors of a catalog.		
Parameters	osid.id.Id	catalogId	the Id of the registered Catalog
	osid.id.Id	ancestorId	the Id of the removed ancestor catalog
Compliance	mandatory		This method must be implemented.
Method	**deletedDescendantCatalog**		
Description	The callback for notifications of deleted descendants of a catalog.		
Parameters	osid.id.Id	catalogId	the Id of the registered Catalog
	osid.id.Id	descendantId	the Id of the deleted descendant catalog
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogList
Implements	osid.OsidList

Description	Like all OsidLists, CatalogList provides a means for accessing Catalog elements sequentially either one at a time or many at a time. Examples:
	
	`while (cl.hasNext()) {`
	`    Catalog catalog = al.getNextCatalog();`
	`}`
	
	or
	
	`while (cl.hasNext()) {`
	`    Catalog[] catalogs = cl.getNextCatalogs(cl.available());`
	`}`

Method	getNextCatalog	
Description	Gets the next Catalog in this list.	
Return	osid.cataloging.Catalog	the next Catalog in this list. The hasNext() method should be used to test that a next Catalog is available before calling this method.
Errors	ILLEGAL_STATE	no more elements available in this list
	OPERATION_FAILED	unable to complete request
Compliance	mandatory	This method must be implemented.

Method	getNextCatalogs		
Description	Gets the next set of Catalog elements in this list which must be less than or equal to the number returned from available().		
Parameters	cardinal	n	the number of Catalog elements requested which should be less than or equal to available()
Return	osid.cataloging.Catalog[]		an array of Catalog elements. The length of the array is less than or equal to the number specified.
Errors	ILLEGAL_STATE		no more elements available in this list
	OPERATION_FAILED		unable to complete request
Compliance	mandatory		This method must be implemented.

Interface	osid.cataloging.CatalogEntryReceiver		
Implements	osid.OsidReceiver		
Description	The catalog receiver is the consumer supplied interface for receiving notifications pertaining to new, updated or deleted Ids in Catalogs.		
Method	**newCatalogEntry**		
Description	The callback for notifications of new catalogs entries.		
Parameters	osid.id.Id	registeredCatalogId	the Id of the registered Catalog
	osid.id.Id	catalogId	the Id of the Catalog that triggered this event. May be the same as the registered catalog or one of its descendants.
	osid.id.Id	entryId	the Id of the new entry
Compliance	mandatory		This method must be implemented.
Method	**deletedCatalogEntry**		
Description	the callback for notification of deleted catalog entries.		
Parameters	osid.id.Id	registeredCatalogId	the Id of the registered Catalog
	osid.id.Id	catalogId	the Id of the Catalog that triggered this event. May be the same as the registered catalog or one of its descendants.
	osid.id.Id	entryId	the Id of the deleted entry
Compliance	mandatory		This method must be implemented.