



## **OSID V3 Specifications filing package**

Version Draft 3

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

Last Modified: 30 October 2008

prepared by:  
Tom Coppeto  
OnTapSolutions

Copyright © 2008 Massachusetts Institute of Technology

	<b>OSID License</b>
<b>Copyright</b>	<p>Copyright © 2008 Massachusetts Institute of Technology. All Rights Reserved.</p> <p>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.</p>
<b>License</b>	<p>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.</p> <p>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.</p> <p>You may modify 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.</p> <p>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 contemplating export to obtain such a license before exporting this Work.</p>

Package Description	osid.filing package
Interfaces	<code>osid.filing.FilingProfile</code> <code>osid.filing.FilingManager</code> <code>osid.filing.FilingProxyManager</code> <code>osid.filing.FileSession</code> <code>osid.filing.FileSearchSession</code> <code>osid.filing.FileNotificationSession</code> <code>osid.filing.DirectoryEntryLookupSession</code> <code>osid.filing.DirectorySearchSession</code> <code>osid.filing.DirectoryAdminSession</code> <code>osid.filing.DirectoryNotificationSession</code> <code>osid.filing.FileManagementSession</code> <code>osid.filing.AllocationSession</code> <code>osid.filing.AllocationAdminSession</code> <code>osid.filing.AllocationNotificationSession</code> <code>osid.filing.DirectoryEntry</code> <code>osid.filing.DirectoryEntryQuery</code> <code>osid.filing.File</code> <code>osid.filing.FileRecord</code> <code>osid.filing.FileQuery</code> <code>osid.filing.FileQueryRecord</code> <code>osid.filing.FileForm</code> <code>osid.filing.FileFormRecord</code> <code>osid.filing.FileSearchOrder</code> <code>osid.filing.FileSearchOrderRecord</code> <code>osid.filing.FileSearch</code> <code>osid.filing.FileSearchRecord</code> <code>osid.filing.FileSearchResults</code> <code>osid.filing.FileSearchResultsRecord</code> <code>osid.filing.FileReceiver</code> <code>osid.filing.FileList</code> <code>osid.filing.Directory</code> <code>osid.filing.DirectoryRecord</code> <code>osid.filing.DirectoryQuery</code> <code>osid.filing.DirectoryQueryRecord</code> <code>osid.filing.DirectoryForm</code> <code>osid.filing.DirectoryFormRecord</code> <code>osid.filing.DirectorySearchOrder</code> <code>osid.filing.DirectorySearchOrderRecord</code> <code>osid.filing.DirectorySearch</code> <code>osid.filing.DirectorySearchRecord</code> <code>osid.filing.DirectorySearchResults</code> <code>osid.filing.DirectorySearchResultsRecord</code> <code>osid.filing.DirectoryReceiver</code> <code>osid.filing.DirectoryList</code> <code>osid.filing.Allocation</code>

osid.filing.AllocationReceiver

<b>Package</b>	<b>osid.filing</b>
<b>Title</b>	<b>Filing Open Service Interface Definitions</b>
<b>Version</b>	3.0.0
<b>Description</b>	<p>The Filing OSID provides a means for managing and accessing files and directories. The Filing OSID is used to abstract assumptions made about using a specific file system, or can be used to provide a file-based application a file system oriented view of other OSIDs.</p> <p>The Filing OSID defines a file access session that maps to a single file. This is the simplest application of the Filing OSID can be used in circumstances when it is desirable to confine knowledge of a particular path within a provider. In visible federation, the consumer may select a file based on a given path name.</p> <p>Unlike the other OSIDs, the identity of a file is based on a path string rather than an Id. The path name serves as a unique identifier. Files also have a name which is used as an identifier with respect to a specific directory. Generally, the name is the last component of the path but the OSIDs don't specify a <u>particular relationship between the file name and its absolute path</u>.</p> <p>The Filing OSID differs from common file system APIs in its separate treatment of files and directories. An application may access a single file, or query for all files and subdirectories within a given directory. Directory browsing is performed through the DirectoryEntryLookupSession. To browse a different directory, a new lookup session is spawned.</p> <p>Files and Directories are not OSID Objects although some of their characteristics are inherited from the general OSID Object pattern. The Filing OSID sits somewhere between a generic transport or persistence service and a more structured OSID object. As such, Files and Directories also do not have extensible interfaces or Types. Corresponding metadata can be handled in sidecar files and incorporated into a higher level interface such as Asset.</p> <p><del>Files and Directories may support searches and notifications in a similar pattern to the higher level OSIDs. However, the hierarchy in the Filing service is implicit involving a navigation of parent/child relationships from within the Directory. The hierarchy is also restricted to having single parents under a single root.</del></p> <p>Sessions are organized into operations on a single file, operations within a single directory, and operations across directories. Since provider boundaries will generally correspond to directory nodes, it is desirable to contain authentication and state management within a single session for most common operations.</p> <p>The naming of the parameters include:</p> <ul style="list-style-type: none"> <li>• name: name of a file or directory without the path component</li> <li>• path: absolute or relative path of the file or directory including the name component</li> <li>• absolute path: absolute path from the root including the name component</li> </ul>

Basic example:

Or, the session may support direct access to specified files.

<b>Interface</b>	<b>osid.filing.FilingProfile</b>	
<b>Implements</b>	<a href="#">osid.OsidProfile</a>	
<b>Description</b>	The filing profile describes the interoperability among filing services.	
<b>Method</b>	<b>supportsVisibleFederation</b>	
<b>Description</b>	Tests if any dictionary federation is exposed. Federation is exposed when a specific dictionary may be identified, selected and used to create a lookup or admin session. Federation is not exposed when a set of dictionaries appears as a single dictionary.	true if federation is visible false otherwise
<b>Return</b>	<a href="#">boolean</a>	true if federation is visible false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsFiles</b>	
<b>Description</b>	Tests if a FileSession is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a FilingSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsFileSearch</b>	
<b>Description</b>	Tests if file searching is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a FilingSearchSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsFileNotification</b>	
<b>Description</b>	Tests if file notification is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a FilingNotificationSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsDirectoryEntryLookup</b>	
<b>Description</b>	Tests if a DirectoryEntryLookupSession is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a DirectoryEntryLookupSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsDirectorySearch</b>	
<b>Description</b>	Tests if directory searching is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a DirectorySearchSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsDirectoryAdmin</b>	
<b>Description</b>	Tests if directory administration is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a DirectoryAdminSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsDirectoryNotification</b>	
<b>Description</b>	Tests if a directory notification service is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a DirectoryNotificationSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>supportsFilingManagement</b>	
<b>Description</b>	Tests if a file management service is supported.	
<b>Return</b>	<a href="#">boolean</a>	true if a FileManagementSession is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Method</b>	<b>supportsAllocation</b>	
<b>Description</b>	Tests if filing allocation is supported.	
<b>Return</b>	boolean	true if a AllocationSession is available, false otherwise
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>supportsAllocationAdmin</b>	
<b>Description</b>	Tests if quota administration is supported.	
<b>Return</b>	boolean	true if a AllocationAdminSession is available, false otherwise
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>supportsAllocationNotification</b>	
<b>Description</b>	Tests if an allocation notification service is supported.	
<b>Return</b>	boolean	true if a AllocationNotificationSession is available, false otherwise
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getFileRecordTypes</b>	
<b>Description</b>	Gets the supported file record types.	
<b>Return</b>	osid.type.TypeList	a list containing the supported File record types
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>supportsFileRecordType</b>	
<b>Description</b>	Tests if the given file record type is supported.	
<b>Parameters</b>	osid.type.Type   fileRecordType	a Type indicating a file record type
<b>Return</b>	boolean	true if the given record Type is supported, false otherwise
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getFileSearchRecordTypes</b>	
<b>Description</b>	Gets the supported file search record types.	
<b>Return</b>	osid.type.TypeList	a list containing the supported File search record types
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>supportsFileSearchRecordType</b>	
<b>Description</b>	Tests if the given file search record type is supported.	
<b>Parameters</b>	osid.type.Type   fileSearchRecordType	a Type indicating a file search record type
<b>Return</b>	boolean	true if the given search record Type is supported, false otherwise
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getDirectoryRecordTypes</b>	
<b>Description</b>	Gets the supported directory record types.	
<b>Return</b>	osid.type.TypeList	a list containing the supported Directory record types
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>supportsDirectoryRecordType</b>	
<b>Description</b>	Tests if the given directory record type is supported.	
<b>Parameters</b>	osid.type.Type   directoryRecordType	a Type indicating a directory record type
<b>Return</b>	boolean	true if the given record Type is supported, false otherwise
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getDirectorySearchRecordTypes</b>	
<b>Description</b>	Gets the supported directory search record types.	
<b>Return</b>	osid.type.TypeList	a list containing the supported Directory search record types
<b>Compliance</b>	mandatory	This method must be implemented.

Method	<b>supportsDirectorySearchRecordType</b>	
<b>Description</b>	Tests if the given directory search record type is supported.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	directorySearchRecordType a Type indicating a directory search record type
<b>Return</b>	<a href="#">boolean</a>	true if the given search record Type is supported, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Interface	osid.filing.FilingManager	
Implements	<a href="#">osid.OsidManager</a> <a href="#">osid.filing.FilingProfile</a>	
Description	<p>The filing manager provides access sessions to retrieve and manage files and directories. A manager may support federation in that files and directories can be accessed by a specified path. The sessions included in this manager are:</p> <ul style="list-style-type: none"> <li>• FileSession: a basic session for reading and writing a file</li> <li>• FileSearchSession: a session for searching for files</li> <li>• FileNotificationSession: a session for subscribing to changes in files</li> <li>• DirectoryEntryLookupSession: a session for examining the contents of a directory</li> <li>• DirectorySearchSession: a session for searching for directories</li> <li>• DirectoryNotificationSession: a session for subscribing to changes in directories</li> <li>• FilingManagementSession: a session for performing operations across directories</li> <li>• AllocationSession: a session for accessing usage information and quotas</li> <li>• AllocationAdminSession a session for assigning quotas</li> <li>• AllocationNotificationSession: a session for subscribing to usage warnings and quota changes</li> </ul>	
Method	<b>getFileSession</b>	
Description	Gets the session for manipulating files.	
Return	<a href="#">osid.filing.FileSession</a>	the FileSession
Errors	<a href="#">OPERATION_FAILED</a> <a href="#">UNIMPLEMENTED</a>	unable to complete request supportsFiles() is false
Compliance	<a href="#">optional</a>	This method must be implemented if supportsFiles() is true.
Method	<b>getFileSessionForPath</b>	
Description	Gets the session for manipulating files for the given path. If a path represents an alias, the target file is accessed. The path will indicate the path to the alias and the real path will indicate the path to the target file.	
Parameters	<a href="#">string</a> filePath	the pathname to the file
Return	<a href="#">osid.filing.FileSession</a>	a FileSession
Errors	<a href="#">INVALID_ARGUMENT</a> <a href="#">NOT_FOUND</a> <a href="#">NULL_ARGUMENT</a> <a href="#">OPERATION_FAILED</a> <a href="#">UNIMPLEMENTED</a>	filePath is not a file filePath is not found filePath is null unable to complete request supportsFiles() is false
Compliance	<a href="#">optional</a>	This method must be implemented if supportsFiles() and supportsVisibleFederation() are true.
Method	<b>getFileSearchSession</b>	
Description	Gets the session for searching for files.	
Return	<a href="#">osid.filing.FileSearchSession</a>	the FileSessionSession
Errors	<a href="#">OPERATION_FAILED</a> <a href="#">UNIMPLEMENTED</a>	unable to complete request supportsFileSearch() is false
Compliance	<a href="#">optional</a>	This method must be implemented if supportsFileSearch() is true.

Method	getFileSearchSessionForDirectory	
Description	Gets a file search session for the specified directory.	
Parameters	string directoryPath	the pathname to the directory
Return	osid.filing.FileSearchSession	a FileSearchSession
Errors	INVALID_ARGUMENT	directoryPath is not a directory or an alias to a
	NOT_FOUND	directoryPath is not found
	NULL_ARGUMENT	directoryPath is null
	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsFileSearch() is false
Compliance	optional	
	This method must be implemented if supportsFileSearch() and supportsVisibleFederation() are true.	
Method	getFileNotificationSession	
Description	Gets the session for receiving messages about changes to files.	
Parameters	osid.filing.FileReceiver receiver	the notification callback
Return	osid.filing.FileNotificationSession	a FileNotificationSession
Errors	NULL_ARGUMENT	receiver is null
	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsFileNotification() is false
Compliance	optional	
	This method must be implemented if supportsFileNotification() is true.	
Method	getFileNotificationSessionForDirectory	
Description	Gets a file notification session for the specified directory.	
Parameters	osid.filing.FileReceiver receiver	the notification callback
	string directoryPath	the pathname to the directory
Return	osid.filing.FileNotificationSession	a FileNotificationSession
Errors	INVALID_ARGUMENT	directoryPath is not a directory or an alias to a directory
	NOT_FOUND	directoryPath is not found
	NULL_ARGUMENT	receiver or directoryPath is null
	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsFileNotification() is false
Compliance	optional	
	This method must be implemented if supportsFileNotification() and supportsVisibleFederation() are true.	
Method	getDirectoryEntryLookupSession	
Description	Gets the session for examining directories.	
Return	osid.filing.DirectoryEntryLookupSession	a DirectoryEntryLookupSession
Errors	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsDirectoryEntryLookup() is false
Compliance	optional	
	This method must be implemented if supportsDirectoryEntryLookup() is true.	
Method	getDirectoryEntryLookupSessionForDirectory	
Description	Gets the session for examining a given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.	
Parameters	string directoryPath	the pathname to the directory
Return	osid.filing.DirectoryEntryLookupSession	a DirectoryEntryLookupSession
Errors	INVALID_ARGUMENT	directoryPath is not a directory or an alias to a directory
	NOT_FOUND	directoryPath is not found
	NULL_ARGUMENT	directoryPath is null
	OPERATION_FAILED	unable to complete request
	UNIMPLEMENTED	supportsDirectoryEntryLookup() is false
Compliance	optional	
	This method must be implemented if supportsDirectoryEntryLookup() and supportsVisibleFederation() are true.	

Method	<b>getDirectorySearchSession</b>	
<b>Description</b>	Gets the session for searching for directories.	
<b>Return</b>	<a href="#">osid.filing.DirectorySearchSession</a>	a DirectorySearchSession
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">UNIMPLEMENTED</a>	supportsDirectorySearch() is false
<b>Compliance</b>	<a href="#">optional</a>	This method must be implemented if supportsDirectorySearch() is true.
Method	<b>getDirectorySearchSessionForDirectory</b>	
<b>Description</b>	Gets the session for searching for directories within a given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.	
<b>Parameters</b>	<a href="#">string</a>   <a href="#">directoryPath</a>	the pathname to the directory
<b>Return</b>	<a href="#">osid.filing.DirectorySearchSession</a>	a DirectorySearchSession
<b>Errors</b>	<a href="#">INVALID_ARGUMENT</a>	directoryPath is not a directory or an alias to a directory
	<a href="#">NOT_FOUND</a>	directoryPath is not found
	<a href="#">NULL_ARGUMENT</a>	directoryPath is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">UNIMPLEMENTED</a>	supportsDirectorySearch() is false
<b>Compliance</b>	<a href="#">optional</a>	This method must be implemented if supportsDirectorySearch() and supportsVisibleFederation() are true.
Method	<b>getDirectoryAdminSession</b>	
<b>Description</b>	Gets the session for creating and removing files.	
<b>Return</b>	<a href="#">osid.filing.DirectoryAdminSession</a>	a DirectoryAdminSession
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">UNIMPLEMENTED</a>	supportsDirectoryAdmin() is false
<b>Compliance</b>	<a href="#">optional</a>	This method must be implemented if supportsDirectoryAdmin() is true.
Method	<b>getDirectoryAdminSessionForDirectory</b>	
<b>Description</b>	Gets the session for searching for creating and removing files in the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.	
<b>Parameters</b>	<a href="#">string</a>   <a href="#">directoryPath</a>	the pathname to the directory
<b>Return</b>	<a href="#">osid.filing.DirectoryAdminSession</a>	a DirectoryAdminSession
<b>Errors</b>	<a href="#">INVALID_ARGUMENT</a>	directoryPath is not a directory or an alias to a directory
	<a href="#">NOT_FOUND</a>	directoryPath is not found
	<a href="#">NULL_ARGUMENT</a>	directoryPath is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">UNIMPLEMENTED</a>	supportsDirectoryAdmin() is false
<b>Compliance</b>	<a href="#">optional</a>	This method must be implemented if supportsDirectoryAdmin() and supportsVisibleFederation() are true.
Method	<b>getDirectoryNotificationSession</b>	
<b>Description</b>	Gets the session for receiving messages about changes to directories.	
<b>Parameters</b>	<a href="#">osid.filing.DirectoryReceiver</a>   <a href="#">receiver</a>	the notification callback
<b>Return</b>	<a href="#">osid.filing.DirectoryNotificationSession</a>	a DirectoryNotificationSession
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	receiver is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">UNIMPLEMENTED</a>	supportsDirectoryNotification() is false
<b>Compliance</b>	<a href="#">optional</a>	This method must be implemented if supportsDirectoryNotification() is true.

Method	getDirectoryNotificationSessionForDirectory		
Description	Gets the session for receiving messages about changes to directories in the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	<code>osid.filing.DirectoryReceiver</code>	receiver	the notification callback
	<code>string</code>	directoryPath	the pathname to the directory
Return	<code>osid.filing.DirectoryNotificationSession</code>		
Errors	<code>INVALID_ARGUMENT</code>		
	<code>NOT_FOUND</code>		
	<code>NULL_ARGUMENT</code>		
	<code>OPERATION_FAILED</code>		
Compliance	<code>UNIMPLEMENTED</code>		
Method	getFilingManagementSession		
Description	Gets a session for manipulating entries across directories.		
Return	<code>osid.filing.FilingManagementSession</code>		
Errors	<code>OPERATION_FAILED</code>		
	<code>UNIMPLEMENTED</code>		
Compliance	<code>optional</code>		
This method must be implemented if supportsFilingManagement() and supportsVisibleFederation() are true.			
Method	getFileManagementSessionForDirectory		
Description	Gets the session for manipulating entries using the given director as the current directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	<code>string</code>	directoryPath	the pathname to the directory
Return	<code>osid.filing.FilingManagementSession</code>		
Errors	<code>INVALID_ARGUMENT</code>		
	<code>NOT_FOUND</code>		
	<code>NULL_ARGUMENT</code>		
	<code>OPERATION_FAILED</code>		
Compliance	<code>UNIMPLEMENTED</code>		
Method	getAllocationSession		
Description	Gets the session for accessing usage and quotas.		
Return	<code>osid.filing.AllocationSession</code>		
Errors	<code>OPERATION_FAILED</code>		
	<code>UNIMPLEMENTED</code>		
Compliance	<code>optional</code>		
This method must be implemented if supportsAllocation() is true.			

Method	getAllocationSessionForDirectory	
Description	Gets the session for accessing usage and quotas for a given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.	
Parameters	string directoryPath	the pathname to the directory
Return	osid.filing.AllocationSession	an AllocationSession
Errors	INVALID_ARGUMENT NOT_FOUND NULL_ARGUMENT OPERATION_FAILED UNIMPLEMENTED	directoryPath is not a directory or an alias to a directory directoryPath is not found directoryPath is null unable to complete request supportsAllocation() is false
Compliance	optional	This method must be implemented if supportsAllocation() and supportsVisibleFederation() are true.
Method	getAllocationAdminSession	
Description	Gets the session for assigning quotas.	
Return	osid.filing.AllocationAdminSession	an AllocationAdminSession
Errors	OPERATION_FAILED UNIMPLEMENTED	unable to complete request supportsAllocationAdmin() is false
Compliance	optional	This method must be implemented if supportsAllocationAdmin() is true.
Method	getAllocationAdminSessionForDirectory	
Description	Gets the session for assigning quotas for the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.	
Parameters	string directoryPath	the pathname to the directory
Return	osid.filing.AllocationAdminSession	an AllocationAdminSession
Errors	INVALID_ARGUMENT NOT_FOUND NULL_ARGUMENT OPERATION_FAILED UNIMPLEMENTED	directoryPath is not a directory or an alias to a directory directoryPath is not found directoryPath is null unable to complete request supportsAllocationAdmin() is false
Compliance	optional	This method must be implemented if supportsAllocationAdmin() and supportsVisibleFederation() are true.
Method	getAllocationNotificationSession	
Description	Gets the session for receiving messages about changes to directories.	
Parameters	osid.filing.AllocationReceiver receiver	the notification callback
Return	osid.filing.AllocationNotificationSession	an AllocationNotificationSession
Errors	NULL_ARGUMENT OPERATION_FAILED UNIMPLEMENTED	receiver is null unable to complete request supportsAllocationNotification() is false
Compliance	optional	This method must be implemented if supportsDirectoryNotification() is true.

Method	<b>getAllocationNotificationSessionForDirectory</b>		
<b>Description</b>	Gets the session for receiving messages about usage warnings and quota changes for the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
<b>Parameters</b>	<code>osid.filing.AllocationReceiver</code>	receiver	the notification callback
	<code>string</code>	directoryPath	the pathname to the directory
<b>Return</b>	<code>osid.filing.AllocationNotificationSession</code>		
	<code>INVALID_ARGUMENT</code>		
	<code>NOT_FOUND</code>		
	<code>NULL_ARGUMENT</code>		
	<code>OPERATION_FAILED</code>		
	<code>UNIMPLEMENTED</code>		
<b>Errors</b>			
<b>Compliance</b>	<code>optional</code>	This method must be implemented if supportsAllocationNotification() and supportsVisibleFederation() are true.	

Interface	osid.filing.FilingProxyManager		
Implements	<a href="#">osid.OsidProxyManager</a> <a href="#">osid_filing.FilingProfile</a>		
Description	<p>The filing manager provides access sessions to retrieve and manage files and directories. A manager may support federation in that files and directories can be accessed by a specified path. Methods in this manager support the passing of an Authentication object for the purposes of proxy authentication. The sessions included in this manager are:</p> <ul style="list-style-type: none"> <li>• FileSession: a basic session for reading and writing a file</li> <li>• FileSearchSession: a session for searching for files</li> <li>• FileNotificationSession: a session for subscribing to changes in files</li> <li>• DirectoryEntryLookupSession: a session for examining the contents of a directory</li> <li>• DirectorySearchSession: a session for searching for directories</li> <li>• DirectoryNotificationSession: a session for subscribing to changes in directories</li> <li>• FilingManagementSession: a session for performing operations across directories</li> <li>• AllocationSession: a session for accessing usage information and quotas</li> <li>• AllocationAdminSession a session for assigning quotas</li> <li>• AllocationNotificationSession: a session for subscribing to usage warnings and quota changes</li> </ul>		
Method	<b>getFileSession</b>		
Description	Gets the session for manipulating files.		
Parameters	<a href="#">osid.authentication.Authentication</a>	authentication	proxy authentication
Return	<a href="#">osid.filing.FileSession</a>		
Errors	<p><b>NULL_ARGUMENT</b>  <b>OPERATION_FAILED</b>  <b>PERMISSION_DENIED</b>  <b>UNIMPLEMENTED</b>  <b>UNSUPPORTED</b></p>		
Compliance	<p><b>optional</b></p> <p>This method must be implemented if supportsFiles() is true.</p>		
Method	<b>getFileSessionForPath</b>		
Description	Gets the session for manipulating files for the given path. If a path represents an alias, the target file is accessed. The path will indicate the path to the alias and the real path will indicate the path to the target file.		
Parameters	<a href="#">string</a>	filePath	the pathname to the file
	<a href="#">osid.authentication.Authentication</a>	authentication	proxy authentication
Return	<a href="#">osid.filing.FileSession</a>		
Errors	<p><b>INVALID_ARGUMENT</b>  <b>NOT_FOUND</b>  <b>NULL_ARGUMENT</b>  <b>OPERATION_FAILED</b>  <b>PERMISSION_DENIED</b>  <b>UNIMPLEMENTED</b>  <b>UNSUPPORTED</b></p>		
Compliance	<p><b>optional</b></p> <p>This method must be implemented if supportsFiles() and supportsVisibleFederation() are true.</p>		

Method	getFileSearchSession		
Description	Gets the session for searching for files.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.filing.FileSearchSession		the FileSessionSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsFileSearch() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsFileSearch() is true.
Method	getFileSearchSessionForDirectory		
Description	Gets a file search session for the specified directory.		
Parameters	string	directoryPath	the pathname to the directory
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.filing.FileSearchSession		a FileSearchSession
Errors	INVALID_ARGUMENT		directoryPath is not a directory or an alias to a directory
	NOT_FOUND		directoryPath is not found
	NULL_ARGUMENT		directoryPath or authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsFileSearch() is false
Errors	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsFileSearch() and supportsVisibleFederation() are true.
Method	getFileNotificationSession		
Description	Gets the session for receiving messages about changes to files.		
Parameters	osid.filing.FileReceiver	receiver	the notification callback
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.filing.FileNotificationSession		a FileNotificationSession
Errors	NULL_ARGUMENT		receiver or authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsFileNotification() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsFileNotification() is true.

Method	getFileNotificationSessionForDirectory		
Description	Gets a file notification session for the specified directory.		
Parameters	<a href="#">osid.filing.FileReceiver</a>	receiver	the notification callback
	<a href="#">string</a>	directoryPath	the pathname to the directory
	<a href="#">osid.authentication.Authentication</a>	authentication	proxy authentication
Return	<a href="#">osid.filing.FileNotificationSession</a>		a FileNotificationSession
Errors	<a href="#">INVALID_ARGUMENT</a>		directoryPath is not a directory or an alias to a directory
	<a href="#">NOT_FOUND</a>		directoryPath is not found
	<a href="#">NULL_ARGUMENT</a>		receiver, directoryPath or authentication is null
	<a href="#">OPERATION_FAILED</a>		unable to complete request
	<a href="#">PERMISSION_DENIED</a>		authentication is invalid
	<a href="#">UNIMPLEMENTED</a>		supportsFileNotification() is false
	<a href="#">UNSUPPORTED</a>		the authentication service is not supported
Compliance	<a href="#">optional</a>		This method must be implemented if supportsFileNotification() and supportsVisibleFederation() are true.
Method	getDirectoryEntryLookupSession		
Description	Gets the session for examining directories.		
Parameters	<a href="#">osid.authentication.Authentication</a>	authentication	proxy authentication
Return	<a href="#">osid.filing.DirectoryEntryLookupSession</a>		a DirectoryEntryLookupSession
Errors	<a href="#">NULL_ARGUMENT</a>		authentication is null
	<a href="#">OPERATION_FAILED</a>		unable to complete request
	<a href="#">PERMISSION_DENIED</a>		authentication is invalid
	<a href="#">UNIMPLEMENTED</a>		supportsDirectoryEntryLookup() is false
	<a href="#">UNSUPPORTED</a>		the authentication service is not supported
Compliance	<a href="#">optional</a>		This method must be implemented if supportsDirectoryEntryLookup() is true.
Method	getDirectoryEntryLookupSessionForDirectory		
Description	Gets the session for examining a given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	<a href="#">string</a>	directoryPath	the pathname to the directory
	<a href="#">osid.authentication.Authentication</a>	authentication	proxy authentication
Return	<a href="#">osid.filing.DirectoryEntryLookupSession</a>		a DirectoryEntryLookupSession
Errors	<a href="#">INVALID_ARGUMENT</a>		directoryPath is not a directory or an alias to a directory
	<a href="#">NOT_FOUND</a>		directoryPath is not found
	<a href="#">NULL_ARGUMENT</a>		directoryPath or authentication is null
	<a href="#">OPERATION_FAILED</a>		unable to complete request
	<a href="#">PERMISSION_DENIED</a>		authentication is invalid
	<a href="#">UNIMPLEMENTED</a>		supportsDirectoryEntryLookup() is false
	<a href="#">UNSUPPORTED</a>		the authentication service is not supported
Compliance	<a href="#">optional</a>		This method must be implemented if supportsDirectoryEntryLookup() and supportsVisibleFederation() are true.

Method	getDirectorySearchSession	
Description	Gets the session for searching for directories.	
Parameters	osid.authentication.Authentication	authentication proxy authentication
Return	osid.filing.DirectorySearchSession	a DirectorySearchSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsDirectorySearch() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsDirectorySearch() is true.
Method	getDirectorySearchSessionForDirectory	
Description	Gets the session for searching for directories within a given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.	
Parameters	string directoryPath	the pathname to the directory
Parameters	osid.authentication.Authentication	authentication proxy authentication
Return	osid.filing.DirectorySearchSession	a DirectorySearchSession
Errors	INVALID_ARGUMENT	directoryPath is not a directory or an alias to a directory
	NOT_FOUND	directoryPath is not found
	NULL_ARGUMENT	directoryPath or authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsDirectorySearch() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsDirectorySearch() and supportsVisibleFederation() are true.
Method	getDirectoryAdminSession	
Description	Gets the session for creating and removing files.	
Parameters	osid.authentication.Authentication	authentication proxy authentication
Return	osid.filing.DirectoryAdminSession	a DirectoryAdminSession
Errors	NULL_ARGUMENT	authentication is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authentication is invalid
	UNIMPLEMENTED	supportsDirectoryAdmin() is false
	UNSUPPORTED	the authentication service is not supported
Compliance	optional	This method must be implemented if supportsDirectoryAdmin() is true.

Method	<b>getDirectoryAdminSessionForDirectory</b>		
Description	Gets the session for searching for creating and removing files in the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	string	directoryPath	the pathname to the directory
osid.authentication.Authentication	authentication	proxy authentication	
Return	<a href="#">osid.filing.DirectoryAdminSession</a>		a DirectoryAdminSession
Errors			
INVALID_ARGUMENT			directoryPath is not a directory or an alias to a directory
NOT_FOUND			directoryPath is not found
NULL_ARGUMENT			directoryPath or authentication is null
OPERATION_FAILED			unable to complete request
PERMISSION_DENIED			authentication is invalid
UNIMPLEMENTED			supportsDirectoryAdmin() is false
UNSUPPORTED			the authentication service is not supported
Compliance	<a href="#">optional</a>		This method must be implemented if supportsDirectoryAdmin() and supportsVisibleFederation() are true.
Method	<b>getDirectoryNotificationSession</b>		
Description	Gets the session for receiving messages about changes to directories.		
Parameters	osid.filing.DirectoryReceiver	receiver	the notification callback
osid.authentication.Authentication	authentication	proxy authentication	
Return	<a href="#">osid.filing.DirectoryNotificationSession</a>		a DirectoryNotificationSession
Errors			
NULL_ARGUMENT			receiver or authentication is null
OPERATION_FAILED			unable to complete request
PERMISSION_DENIED			authentication is invalid
UNIMPLEMENTED			supportsDirectoryNotification() is false
UNSUPPORTED			the authentication service is not supported
Compliance	<a href="#">optional</a>		This method must be implemented if supportsDirectoryNotification() is true.
Method	<b>getDirectoryNotificationSessionForDirectory</b>		
Description	Gets the session for receiving messages about changes to directories in the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	osid.filing.DirectoryReceiver	receiver	the notification callback
string	directoryPath	the pathname to the directory	
osid.authentication.Authentication	authentication	proxy authentication	
Return	<a href="#">osid.filing.DirectoryNotificationSession</a>		a DirectoryNotificationSession
Errors			
INVALID_ARGUMENT			directoryPath is not a directory or an alias to a directory
NOT_FOUND			directoryPath is not found
NULL_ARGUMENT			receiver, directoryPath or authentication is null
OPERATION_FAILED			unable to complete request
PERMISSION_DENIED			authentication is invalid
UNIMPLEMENTED			supportsDirectoryNotification() is false
UNSUPPORTED			the authentication service is not supported
Compliance	<a href="#">optional</a>		This method must be implemented if supportsDirectoryNotification() and supportsVisibleFederation() are true.

Method	getFilingManagementSession		
Description	Gets a session for manipulating entries across directories.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.filing.FilingManagementSession		a FilingManagementSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsFilingManagement() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsFilingManagement() is true.
Method	getFileManagementSessionForDirectory		
Description	Gets the session for manipulating entries using the given director as the current directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	string	directoryPath	the pathname to the directory
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.filing.FilingManagementSession		a FilingManagementSession
Errors	INVALID_ARGUMENT		directoryPath is not a directory or an alias to a directory
	NOT_FOUND		directoryPath is not found
	NULL_ARGUMENT		directoryPath or authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsFilingManagement() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsFilingManagement() and supportsVisibleFederation() are true.
Method	getAllocationSession		
Description	Gets the session for accessing usage and quotas.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.filing.AllocationSession		an AllocationSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsAllocation() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsAllocation() is true.

Method	getAllocationSessionForDirectory		
Description	Gets the session for accessing usage and quotas for a given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	string osid.authentication.Authentication	directoryPath authentication	the pathname to the directory proxy authentication
Return	<a href="#">osid.filing.AllocationSession</a>		an AllocationSession
Errors	INVALID_ARGUMENT		directoryPath is not a directory or an alias to a directory
	NOT_FOUND		directoryPath is not found
	NULL_ARGUMENT		directoryPath or authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsAllocation() is false
Compliance	optional		This method must be implemented if supportsAllocation() and supportsVisibleFederation() are true.
Method	getAllocationAdminSession		
Description	Gets the session for assigning quotas.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	<a href="#">osid.filing.AllocationAdminSession</a>		an AllocationAdminSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsAllocationAdmin() is false
Compliance	optional		This method must be implemented if supportsAllocationAdmin() is true.
Method	getAllocationAdminSessionForDirectory		
Description	Gets the session for assigning quotas for the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	string osid.authentication.Authentication	directoryPath authentication	the pathname to the directory proxy authentication
Return	<a href="#">osid.filing.AllocationAdminSession</a>		an AllocationAdminSession
Errors	INVALID_ARGUMENT		directoryPath is not a directory or an alias to a directory
	NOT_FOUND		directoryPath is not found
	NULL_ARGUMENT		directoryPath or authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsAllocationAdmin() is false
Compliance	optional		This method must be implemented if supportsAllocationAdmin() and supportsVisibleFederation() are true.
Method	getAllocationNotificationSession		
Description	Gets the session for receiving messages about changes to directories.		
Parameters	osid.filing.AllocationReceiver	receiver	the notification callback
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	<a href="#">osid.filing.AllocationNotificationSession</a>		an AllocationNotificationSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		authentication is invalid
	UNIMPLEMENTED		supportsAllocationNotification() is false
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsDirectoryNotification() is true.

Method	getAllocationNotificationSessionForDirectory		
Description	Gets the session for receiving messages about usage warnings and quota changes for the given directory. If the path is an alias, the target directory is used. The path indicates the file alias and the real path indicates the target directory.		
Parameters	<code>osid.filing.AllocationReceiver</code>	receiver	the notification callback
	<code>string</code>	directoryPath	the pathname to the directory
	<code>osid.authentication.Authentication</code>	authentication	proxy authentication
Return	<code>osid.filing.AllocationNotificationSession</code>		an AllocationNotificationSession
Errors	<code>INVALID_ARGUMENT</code>		directoryPath is not a directory or an alias to a directory
	<code>NOT_FOUND</code>		directoryPath is not found
	<code>NULL_ARGUMENT</code>		receiver, directoryPath or authentication is null
	<code>OPERATION_FAILED</code>		unable to complete request
	<code>PERMISSION_DENIED</code>		authentication is invalid
	<code>UNIMPLEMENTED</code>		<code>supportsAllocationNotification()</code> is false
Compliance	<code>UNSUPPORTED</code>		the authentication service is not supported
	<code>optional</code>		This method must be implemented if <code>supportsAllocationNotification()</code> and <code>supportsVisibleFederation()</code> are true.

<b>Interface</b>	<b>osid.filing.FileSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	This session is for reading and writing the file associated with this session.	
<b>Method</b>	<b>getFilePath</b>	
<b>Description</b>	Gets the absolute path of this file.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this file
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getFile</b>	
<b>Description</b>	Gets the file associated with this session.	
<b>Return</b>	<a href="#">osid.filing.File</a>	the file associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canReadFile</b>	
<b>Description</b>	Tests if this user can access this file. A return of true does not guarantee successful authorization. A return of false indicates that it is known accessing this file will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer read operations to unauthorized users.	
<b>Return</b>	<a href="#">boolean</a>	false if file access is not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getInputStream</b>	
<b>Description</b>	Gets the input stream for reading this file. The input stream reads until the end of the file.	
<b>Return</b>	<a href="#">osid.transport.DataInputStream</a>	the input stream for reading this file
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getBlockingInputStream</b>	
<b>Description</b>	Gets the input stream for reading this file. The returned input stream, once it reaches the end of the file, blocks for new content that may be later appended to the file.	
<b>Return</b>	<a href="#">osid.transport.DataInputStream</a>	the input stream for reading this file
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canWriteFile</b>	
<b>Description</b>	Tests if this user can update this file. A return of true does not guarantee successful authorization. A return of false indicates that it is known writing to this file will result in a PERMISSION_DENIED. This is intended as a hint to an application that may not wish to offer write operations to unauthorized users.	
<b>Return</b>	<a href="#">boolean</a>	false if file write is not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getOutputStream</b>	
<b>Description</b>	Gets an output stream for writing to this file, replacing any existing contents.	
<b>Return</b>	<a href="#">osid.transport.DataOutputStream</a>	the output stream for writing to this file
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	<b>getOutputStreamForAppend</b>	
<b>Description</b>	Gets an output stream for appending to this file.	
<b>Return</b>	<a href="#">osid.transport.DataOutputStream</a>	the output stream for appending to this file
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>touch</b>	
<b>Description</b>	Updates the modified time of a file to be the current time.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileSearchSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	<p>This session provides methods for searching among files and directories objects. The search query is constructed using the FileQuery and DirectoryQuery interfaces.</p> <p>getFilesByQuery() is the basic search method and returns a list of Files. A more advanced search may be performed with getFilesBySearch(). It accepts a FileSearch interface in addition to the query interface for the purpose of specifying additional options affecting the entire search, such as ordering. getFilesBySearch() returns an FileSearchResults interface that can be used to access the resulting fileList or be used to perform a search within the result set through FileSearch.</p> <p>This session defines views that offer differing behaviors when retrieving multiple objects.</p> <ul style="list-style-type: none"> <li>• federated directory view: searches include entries in directories of which this directory is an ancestor</li> <li>• isolated directory view: searches are restricted to this directory only</li> </ul>	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canSearchFiles</b>	
<b>Description</b>	Tests if this user can perform File 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.	
<b>Return</b>	<a href="#">boolean</a>	false if lookup methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>useFederatedDirectoryView</b>	
<b>Description</b>	Federates the view for methods in this session. A federated view will include entries in directories which are children of this directory.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>useIsolatedDirectoryView</b>	
<b>Description</b>	Isolates the view for methods in this session. An isolated view restricts lookups to this directory only.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getFileQuery</b>	
<b>Description</b>	Gets a file query interface.	
<b>Return</b>	<a href="#">osid.filing.FileQuery</a>	the file query
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	getFilesByQuery	
<b>Description</b>	Gets a list of Files matching the given query.	
<b>Parameters</b>	<a href="#">osid.filing.FileQuery</a> fileQuery	the search query
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the returned FileList
<b>Errors</b>	NULL_ARGUMENT OPERATION_FAILED PERMISSION_DENIED UNSUPPORTED	fileQuery is null unable to complete request authorization failure fileQuery is not of this service
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	getFileSearch	
<b>Description</b>	Gets a file search interface. The returned query only makes available the core FileSearch interface and does not support additional interface types. getFileSearchForInterfaceType() should be used if additional interface types are required.	
<b>Return</b>	<a href="#">osid.filing.FileSearch</a>	the file search interface
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	getFileSearchOrder	
<b>Description</b>	Gets a file search order interface. The FileSearchOrder is supplied to a FileSearch to specify the ordering of results.	
<b>Return</b>	<a href="#">osid.filing.FileSearchOrder</a>	the file search order interface
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	getFilesBySearch	
<b>Description</b>	Gets the search results matching the given search query using the given search interface.	
<b>Parameters</b>	<a href="#">osid.filing.FileQuery</a> fileQuery	the search query
	<a href="#">osid.filing.FileSearch</a> fileSearch	the search interface
<b>Return</b>	<a href="#">osid.filing.FileSearchResults</a>	the search results
<b>Errors</b>	NULL_ARGUMENT OPERATION_FAILED PERMISSION_DENIED UNSUPPORTED	fileQuery or fileSearch is null unable to complete request authorization failure fileQuery or fileSearch is not of this service
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileNotificationSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	<p>This session defines methods to receive notifications on adds/changes to Files. 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.</p> <p>Two view are defined in this session:</p> <ul style="list-style-type: none"> <li>federated view: includes notifications of files located in subdirectories in this directory</li> <li>isolated view: includes notifications of files in this directory only</li> </ul>	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the Directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the Directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canRegisterForFileNotifications</b>	
<b>Description</b>	Tests if this user can register for File 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.	
<b>Return</b>	<a href="#">boolean</a>	false if notification methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>useFederatedDirectoryView</b>	
<b>Description</b>	Federates the view for methods in this session. A federated view will include files in directories which are children of this directory.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>useIsolatedDirectoryView</b>	
<b>Description</b>	Isolates the view for methods in this session. An isolated view restricts notifications to this directory only.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>registerForNewFiles</b>	
<b>Description</b>	Register for notifications of new files. FileReceiver.newFile() is invoked when a new File is created.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>registerForChangedFiles</b>	
<b>Description</b>	Registers for notification of updated files. FileReceiver.changedFile() is invoked when a file is changed.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	<b>registerForChangedFile</b>				
<b>Description</b>	Registers for notification of an updated file. FileReceiver.changedFile() is invoked when the specified file is changed.				
<b>Parameters</b>	<code>string</code>	<code>name</code>	the name of the file		
<b>Errors</b>	<code>NOT_FOUND</code>		a file was not found identified by the given name		
	<code>NULL_ARGUMENT</code>		name is null		
	<code>OPERATION_FAILED</code>		unable to complete request		
	<code>PERMISSION_DENIED</code>		authorization failure		
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.			
Method	<b>registerForDeletedFiles</b>				
<b>Description</b>	Registers for notification of deleted files. FileReceiver.deletedFile() is invoked when a file is deleted.				
<b>Errors</b>	<code>OPERATION_FAILED</code>		unable to complete request		
	<code>PERMISSION_DENIED</code>		authorization failure		
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.			
Method	<b>registerForDeletedFile</b>				
<b>Description</b>	Registers for notification of a deleted file. FileReceiver.changedFile() is invoked when the specified file is changed.				
<b>Parameters</b>	<code>string</code>	<code>name</code>	the name of the file		
<b>Errors</b>	<code>NOT_FOUND</code>		a file was not found identified by the given name		
	<code>NULL_ARGUMENT</code>		name is null		
	<code>OPERATION_FAILED</code>		unable to complete request		
	<code>PERMISSION_DENIED</code>		authorization failure		
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.			

<b>Interface</b>	<b>osid.filing.DirectoryEntryLookupSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	<p>This session defines methods for operating on files and directories. A DirectoryEntrySession is associated with a directory that behaves as the current directory path for supplying relative path names. Absolute path names can be supplied to access any file or directory in the file system.</p> <p>This session defines the following views:</p> <ul style="list-style-type: none"> <li>comparative view: elements may be silently omitted or re-ordered</li> <li>plenary view: provides a complete result set or is an error condition</li> <li>federated directory view: searches include entries in directories of which this directory is an ancestor</li> <li>isolated directory view: lookups are restricted to entries in this directory only</li> </ul> <p>Generally, the comparative view should be used for most applications as it permits operation even if there is data out of sync. 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 precision using the plenary view.</p>	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canLookupDirectoryEntries</b>	
<b>Description</b>	Tests if this user can perform entry 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.	
<b>Return</b>	<a href="#">boolean</a>	false if lookup methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>hasParentDirectory</b>	
<b>Description</b>	Tests if the directory associated with this session has a parent directory.	
<b>Return</b>	<a href="#">boolean</a>	true if a parent exists, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>getParentDirectory</b>	
<b>Description</b>	Gets the parent of the directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the parent of the directory associated with this session
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a>	hasParentDirectory() is false
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>useComparativeDirectoryView</b>	
<b>Description</b>	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.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.

Method	usePlenaryDirectoryView				
Description	A complete view of the file or directory 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	useFederatedDirectoryView				
Description	Federates the view for methods in this session. A federated view will include entries in directories which are children of this directory.				
Compliance	mandatory	This method is must be implemented.			
Method	useIsolatedDirectoryView				
Description	Isolates the view for methods in this session. An isolated view restricts lookups to this directory only.				
Compliance	mandatory	This method is must be implemented.			
Method	exists				
Description	Tests if a file, directory or alias name exists. In a federated view, the existence test is performed on this directory and any children of this directory. In an isolated view, the existence test is restricted to this directory only.				
Parameters	string	name	a file or directory name		
Return	boolean		true if the name exists, false otherwise		
Compliance	mandatory	This method is must be implemented.			
Method	isFile				
Description	Tests if a name exists and is a file or an alias to a file. In a federated view, the existence test is performed on this directory and any children of this directory. In an isolated view, the existence test is restricted to this directory only.				
Parameters	string	name	a file name		
Return	boolean		true if the name is a file, false otherwise		
Compliance	mandatory	This method is must be implemented.			
Method	isDirectory				
Description	Tests if a name exists and is a directory or an alias to a directory. In a federated view, the existence test is performed on this directory and any children of this directory. In an isolated view, the existence test is restricted to this directory only.				
Parameters	string	name	a file or directory name		
Return	boolean		true if the path is a directory, false otherwise		
Compliance	mandatory	This method is must be implemented.			
Method	isAlias				
Description	Tests if a name exists and is an alias. In a federated view, the existence test is performed on this directory and any children of this directory. In an isolated view, the existence test is restricted to this directory only.				
Parameters	string	name	a file or directory name		
Return	boolean		true if the path is an alias, false otherwise		
Compliance	mandatory	This method is must be implemented.			
Method	getFile				
Description	Gets a specified file or alias to the file by its name in the current directory only. For federated views, use getFilesByName().				
Parameters	string	name	the name to the file		
Return	osid.filing.File		the file		
Errors	NOT_FOUND	name is not found or is a directory			
	NULL_ARGUMENT	name is null			
	OPERATION_FAILED	unable to complete request			
	PERMISSION_DENIED	authorization failure			
Compliance	mandatory	This method must be implemented.			

Method	<b>getFiles</b>	
<b>Description</b>	Gets the list of files and aliases to files in this directory. In a federated view, this method returns all files in descendant directories.	
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the list of files in this directory
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getFilesByName</b>	
<b>Description</b>	Gets a specified files and aliases to files for the given file name. In an isolated view, this method behaves like getFile(). Is a federated view, this method returns a list of files by the same name in descendant directories.	
<b>Parameters</b>	<a href="#">string</a>	name
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the list of files of the given name
<b>Errors</b>	<a href="#">NOT_FOUND</a>	name is not found or is a directory
	<a href="#">NULL_ARGUMENT</a>	name is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getFilesByGenusType</b>	
<b>Description</b>	Gets a FileList corresponding to the given file genus Type which does not include files of genus types derived from the specified Type. In plenary mode, the returned list contains all known files or an error results. Otherwise, the returned list may contain only those files that are accessible through this session. In both cases, the order of the set is not specified.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileGenusType
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the returned File list
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileGenusType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getFilesByParentGenusType</b>	
<b>Description</b>	Gets a FileList corresponding to the given file genus Type and include any additional files with genus types derived from the specified Type. In plenary mode, the returned list contains all known files or an error results. Otherwise, the returned list may contain only those files that are accessible through this session. In both cases, the order of the set is not specified.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileGenusType
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the returned File list
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileGenusType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getFilesByRecordType</b>	
<b>Description</b>	Gets a FileList corresponding to the given file record Type. The set of files implementing the given interface type is returned. In plenary mode, the returned list contains all known files or an error results. Otherwise, the returned list may contain only those files that are accessible through this session. In both cases, the order of the set is not specified.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileInterfaceType
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the returned File list
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileInterfaceType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	getSubdirectory				
Description	Gets a specified directory or alias to the directory by its name in the current directory only. For federated views, use getDirectoriesByName().				
Parameters	string	name	the name of the directory		
Return	<a href="#">osid.filing.Directory</a>				
Errors	NOT_FOUND	name is not found or is a file			
	NULL_ARGUMENT	name is null			
	OPERATION_FAILED	unable to complete request			
	PERMISSION_DENIED	authorization failure			
Compliance	mandatory	This method must be implemented.			
Method	getSubdirectories				
Description	Gets the list of directories and aliases to directories in this directory. In a federated view, this method returns all directories in descedent directories.				
Return	<a href="#">osid.filing.DirectoryList</a>				
Errors	OPERATION_FAILED	unable to complete request			
	PERMISSION_DENIED	authorization failure			
Compliance	mandatory	This method must be implemented.			
Method	getSubdirectoriesByName				
Description	Gets a specified directories and aliases to directories for the given directory name. In an isolated view, this method behaves like getDirectory(). Is a federated view, this method returns a list of directories by the same name in descedent directories.				
Parameters	string	name	the name of the file		
Return	<a href="#">osid.filing.FileList</a>				
Errors	NOT_FOUND	name is not found or is a directory			
	NULL_ARGUMENT	name is null			
	OPERATION_FAILED	unable to complete request			
	PERMISSION_DENIED	authorization failure			
Compliance	mandatory	This method must be implemented.			
Method	getSubdirectoriesByGenusType				
Description	Gets a DirectoryList corresponding to the given directory genus Type which does not include directories of genus types derived from the specified Type. In plenary mode, the returned list contains all known directories or an error results. Otherwise, the returned list may contain only those directories that are accessible through this session. In both cases, the order of the set is not specified.				
Parameters	<a href="#">osid.type.Type</a>	directoryGenusType	a directory genus type		
Return	<a href="#">osid.filing.DirectoryList</a>				
Errors	NULL_ARGUMENT	directoryGenusType is null			
	OPERATION_FAILED	unable to complete request			
	PERMISSION_DENIED	authorization failure			
Compliance	mandatory	This method must be implemented.			
Method	getSubdirectoriesByParentGenusType				
Description	Gets a DirectoryList corresponding to the given directory genus Type and include any additional directories with genus types derived from the specified Type. In plenary mode, the returned list contains all known directories or an error results. Otherwise, the returned list may contain only those directories that are accessible through this session. In both cases, the order of the set is not specified.				
Parameters	<a href="#">osid.type.Type</a>	directoryGenusType	a directory genus type		
Return	<a href="#">osid.filing.DirectoryList</a>				
Errors	NULL_ARGUMENT	directoryGenusType is null			
	OPERATION_FAILED	unable to complete request			
	PERMISSION_DENIED	authorization failure			
Compliance	mandatory	This method must be implemented.			

Method	<b>getSubdirectoriesByRecordType</b>	
<b>Description</b>	Gets a DirectoryList corresponding to the given directory record Type. The set of directories implementing the given directory record type is returned. In plenary mode, the returned list contains all known directories or an error results. Otherwise, the returned list may contain only those directories that are accessible through this session. In both cases, the order of the set is not specified.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	directoryRecordType a directory record type
<b>Return</b>	<a href="#">osid.filing.DirectoryList</a>	the returned Directory list
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a> <a href="#">OPERATION_FAILED</a> <a href="#">PERMISSION_DENIED</a>	directoryRecordType is null unable to complete request authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectorySearchSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	<p>This session provides methods for searching among Directory objects. The search query is constructed using the DirectoryQuery interface.</p> <p>This session defines views that offer differing behaviors for searching.</p> <ul style="list-style-type: none"> <li>• federated directory view: searches include directories in directories of which this directory is an ancestor</li> <li>• isolated directory view: searches are restricted to directories in this directory only</li> </ul> <p>getDirectoriesByQuery() is the basic search method and returns a list of Directories. A more advanced search may be performed with getDirectoriesBySearch(). It accepts a DirectorySearch interface in addition to the query interface for the purpose of specifying additional options affecting the entire search, such as ordering. getDirectoriesBySearch() returns an DirectorySearchResults interface that can be used to access the resulting DirectoryList or be used to perform a search within the result set through DirectorySearch.</p>	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canSearchDirectories</b>	
<b>Description</b>	Tests if this user can perform Directory 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.	
<b>Return</b>	<a href="#">boolean</a>	false if search methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>useFederatedDirectoryView</b>	
<b>Description</b>	Federates the view for methods in this session. A federated view will include entries in directories which are children of this directory.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>useIsolatedDirectoryView</b>	
<b>Description</b>	Isolates the view for methods in this session. An isolated view restricts lookups to this directory only.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>getDirectoryQuery</b>	
<b>Description</b>	Gets a directory query interface.	
<b>Return</b>	<a href="#">osid.filing.DirectoryQuery</a>	the directory query
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	<b>getDirectoriesByQuery</b>	
<b>Description</b>	Gets a list of Directory objects matching the given query.	
<b>Parameters</b>	<a href="#">osid.filing.DirectoryQuery</a>	directoryQuery the search query
<b>Return</b>	<a href="#">osid.filing.DirectoryList</a>	the returned DirectoryList
<b>Errors</b>	NULL_ARGUMENT	directoryQuery is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	directoryQuery is not of this service
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getDirectorySearch</b>	
<b>Description</b>	Gets a directory search interface.	
<b>Return</b>	<a href="#">osid.filing.DirectorySearch</a>	the directory search interface
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getDirectorySearchOrder</b>	
<b>Description</b>	Gets a directory search order interface. The DirectorySearchOrder is supplied to a DirectorySearch to specify the ordering of results.	
<b>Return</b>	<a href="#">osid.filing.DirectorySearchOrder</a>	the directory search order interface
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getDirectoriesBySearch</b>	
<b>Description</b>	Gets the search results matching the given search interface.	
<b>Parameters</b>	<a href="#">osid.filing.DirectoryQuery</a>	directoryQuery the search query
	<a href="#">osid.filing.DirectorySearch</a>	directorySearch the search interface
<b>Return</b>	<a href="#">osid.filing.DirectorySearchResults</a>	the search results
<b>Errors</b>	NULL_ARGUMENT	directoryQuery or directorySearch is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	directoryQuery or directorySearch is not of this service
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Interface	osid.filing.DirectoryAdminSession	
Implements	<a href="#">osid.OsidSession</a>	
Description	<p>This session creates and removes files and directories under the directory associated with this session.</p> <p>The view of the administrative methods defined in this session is determined by the provider. For an instance of this session where no repository has been specified, it may not be parallel to the DirectoryLookupSession. For example, a default DirectoryLookupSession may view the entire directory hierarchy while the default DirectoryAdminSession uses an isolated Directory to create new entries. Another scenario is a federated provider who does not wish to permit administrative operations for the federation unaware.</p>	
Method	<b>getDirectoryPath</b>	
Description	Gets the absolute path of this directory.	
Return	<a href="#">string</a>	the absolute path of this directory
Compliance	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getDirectory</b>	
Description	Gets the directory associated with this session.	
Return	<a href="#">osid.filing.Directory</a>	the directory associated with this session
Errors	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
Compliance	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>canModifyDirectory</b>	
Description	Tests if this user can create or remove entries in this directory. A return of true does not guarantee successful authorization. A return of false indicates that it is known modifying this directory will result in a PERMISSION_DENIED. This is intended as a hint to an application that may opt not to offer create operations to an unauthorized user.	
Return	<a href="#">boolean</a>	false if modifying this directory is not authorized, true otherwise
Compliance	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>overwrite</b>	
Description	Overwrite files if a destination pathname exists and is a file.	
Compliance	<a href="#">mandatory</a>	This method is must be implemented.
Method	<b>canCreateFileWithRecordTypes</b>	
Description	Tests if this user can create a single File using the desired record types. While FilingManager.getFileRecordTypes() can be used to examine which records are supported, this method tests which record(s) are required for creating a specific File. Providing an empty array tests if a File can be created with no records.	
Parameters	<a href="#">osid.type.Type[]</a>	fileRecordTypes array of types
Return	<a href="#">boolean</a>	true if File creation using the specified interface Types is supported, false otherwise
Errors	<a href="#">NULL_ARGUMENT</a>	fileRecordTypes is null
Compliance	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getFileFormForCreate</b>	
Description	Gets the file form for creating new files. A new form should be requested for each create transaction.	
Parameters	<a href="#">osid.type.Type</a>	fileInterfaceType the interface Type of the Directory to be created
Return	<a href="#">osid.filing.FileForm</a>	the file form
Errors	<a href="#">NULL_ARGUMENT</a>	fileInterfaceType is null
	<a href="#">UNSUPPORTED</a>	canCreateFileWithRecordTypes(fileInterfaceType) is false
Compliance	<a href="#">mandatory</a>	This method must be implemented.

Method	createFile	
<b>Description</b>	Creates a new file in this directory.	
<b>Parameters</b>	osid.filing.FileForm	fileForm
<b>Return</b>	osid.filing.File	the new file
<b>Errors</b>	ALREADY_EXISTS	name already exists as a file in this directory and overwrite() is false
	INVALID_ARGUMENT	one or more of the form elements is invalid
	NULL_ARGUMENT	fileForm is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	fileForm is not supported
<b>Compliance</b>	mandatory	This method must be implemented.
Method	getFileManagerForUpdate	
<b>Description</b>	Gets the file form for updating an existing files. A new file form should be requested for each update transaction.	
<b>Parameters</b>	string	name
<b>Return</b>	osid.filing.FileForm	the file form
<b>Errors</b>	NOT_FOUND	name is not found
	NULL_ARGUMENT	name is null
	OPERATION_FAILED	unable to complete request
<b>Compliance</b>	mandatory	This method must be implemented.
Method	updateFile	
<b>Description</b>	Updates an existing file.	
<b>Parameters</b>	string	name
	osid.filing.FileForm	fileForm
<b>Errors</b>	INVALID_ARGUMENT	the form contains an invalid value
	NOT_FOUND	name is not found
	NULL_ARGUMENT	name or fileForm is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
	UNSUPPORTED	fileForm is not supported
<b>Compliance</b>	mandatory	This method must be implemented.
Method	removeFile	
<b>Description</b>	Deletes a file.	
<b>Parameters</b>	string	name
<b>Errors</b>	NOT_FOUND	name is not found
	NULL_ARGUMENT	name is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure
<b>Compliance</b>	mandatory	This method must be implemented.
Method	canCreateDirectoryWithRecordTypes	
<b>Description</b>	Tests if this user can create a single Directory using the desired record types. While FilingManager.getDictionaryRecordTypes() can be used to examine which records are supported, this method tests which record(s) are required for creating a specific Directory. Providing an empty array tests if a Directory can be created with no records.	
<b>Parameters</b>	osid.type.Type[]	directoryRecordTypes
<b>Return</b>	boolean	true if Directory creation using the specified record Types is supported, false otherwise
<b>Errors</b>	NULL_ARGUMENT	directoryRecordTypes is null
<b>Compliance</b>	mandatory	This method must be implemented.

Method	<b>getDirectoryFormForCreate</b>	
<b>Description</b>	Gets the file form for creating new directories. A new form should be requested for each create transaction. This method is used for creating new Directories where only the Directory Type is known.	
<b>Parameters</b>	<code>osid.type.Type</code>	directoryInterfaceType the interface Type of the Directory to be created
<b>Return</b>	<code>osid.filing.DirectoryForm</code>	the directory form
<b>Errors</b>	<code>NULL_ARGUMENT</code>	directoryInterfaceType is null
	<code>UNSUPPORTED</code>	canCreateDirectoryWithInterfaceTypes(directoryInterfaceType) is false
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.
Method	<b>createDirectory</b>	
<b>Description</b>	Creates a new directory.	
<b>Parameters</b>	<code>osid.filing.DirectoryForm</code>	directoryForm the directory form
<b>Return</b>	<code>osid.filing.Directory</code>	the new directory
<b>Errors</b>	<code>ALREADY_EXISTS</code>	name already exists in this directory
	<code>INVALID_ARGUMENT</code>	one or more of the form elements is invalid
	<code>NULL_ARGUMENT</code>	directoryForm is null
	<code>OPERATION_FAILED</code>	unable to complete request
	<code>PERMISSION_DENIED</code>	authorization failure
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.
Method	<b>getDirectoryFormForUpdate</b>	
<b>Description</b>	Gets the directory form for updating an existing files. A new directory form should be requested for each update transaction. This method is used when the Directory to be updated is known and is desired to access any metadata specific to the Directory being updated.	
<b>Parameters</b>	<code>string</code>	name name of the directory to update
<b>Return</b>	<code>osid.filing.DirectoryForm</code>	the directory form
<b>Errors</b>	<code>NOT_FOUND</code>	name is not found
	<code>NULL_ARGUMENT</code>	name is null
	<code>OPERATION_FAILED</code>	unable to complete request
	<code>PERMISSION_DENIED</code>	authorization failure
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.
Method	<b>updateDirectory</b>	
<b>Description</b>	Updates an existing directory.	
<b>Parameters</b>	<code>string</code>	name name of the file
	<code>osid.filing.DirectoryForm</code>	directory the form containing the elements to be updated
<b>Errors</b>	<code>INVALID_ARGUMENT</code>	the form contains an invalid value
	<code>NOT_FOUND</code>	name is not found
	<code>NULL_ARGUMENT</code>	name or directoryForm is null
	<code>OPERATION_FAILED</code>	unable to complete request
	<code>PERMISSION_DENIED</code>	authorization failure
	<code>UNSUPPORTED</code>	directoryForm is not supported
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.
Method	<b>removeDirectory</b>	
<b>Description</b>	Deletes a directory in this directory. The directory to remove must be empty.	
<b>Parameters</b>	<code>string</code>	name the name of the directory to delete
<b>Errors</b>	<code>NOT_FOUND</code>	name is not found
	<code>NULL_ARGUMENT</code>	name is null
	<code>OPERATION_FAILED</code>	unable to complete request
	<code>PERMISSION_DENIED</code>	authorization failure
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryNotificationSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	<p>This session defines methods to receive notifications on adds/changes to Directories. 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.</p> <p>Two view are defined in this session:</p> <ul style="list-style-type: none"> <li>federated view: includes notifications of directories located in descendants of this directory</li> <li>isolated view: includes notifications of directories in this directory only</li> </ul>	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the Directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the Directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canRegisterForDirectoryNotifications</b>	
<b>Description</b>	Tests if this user can register for Directory 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.	
<b>Return</b>	<a href="#">boolean</a>	false if notification methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>useFederatedDirectoryView</b>	
<b>Description</b>	Federates the view for methods in this session. A federated view will include directories in directories which are children of this directory.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>useIsolatedDirectoryView</b>	
<b>Description</b>	Isolates the view for methods in this session. An isolated view restricts notifications to this directory only.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method is must be implemented.
<b>Method</b>	<b>registerForNewDirectories</b>	
<b>Description</b>	Register for notifications of new directories. DirectoryReceiver.newDirectory() is invoked when a new Directory is created.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>registerForChangedDirectories</b>	
<b>Description</b>	Registers for notification of updated directories. DirectoryReceiver.changedDirectory() is invoked when a file is changed.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	<b>registerForChangedDirectory</b>				
<b>Description</b>	Registers for notification of an updated file. DirectoryReceiver.changedDirectory() is invoked when the specified directory is changed.				
<b>Parameters</b>	<code>string</code>	<code>name</code>	the name of the directory to monitor		
<b>Errors</b>	<code>NOT_FOUND</code>		a directory was not found identified by the given name		
	<code>NULL_ARGUMENT</code>		name is null		
	<code>OPERATION_FAILED</code>		unable to complete request		
	<code>PERMISSION_DENIED</code>		authorization failure		
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.			
Method	<b>registerForDeletedDirectories</b>				
<b>Description</b>	Registers for notification of deleted directories. DirectoryReceiver.deletedDirectory() is invoked when a directory is deleted.				
<b>Errors</b>	<code>OPERATION_FAILED</code>		unable to complete request		
	<code>PERMISSION_DENIED</code>		authorization failure		
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.			
Method	<b>registerForDeletedDirectory</b>				
<b>Description</b>	Registers for notification of a deleted directory. DirectoryReceiver.changedDirectory() is invoked when the specified directory is changed.				
<b>Parameters</b>	<code>string</code>	<code>name</code>	the name of the directory		
<b>Errors</b>	<code>NOT_FOUND</code>		a directory was not found identified by the given path		
	<code>NULL_ARGUMENT</code>		path is null		
	<code>OPERATION_FAILED</code>		unable to complete request		
	<code>PERMISSION_DENIED</code>		authorization failure		
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.			

Interface	osid.filing.FilingManagementSession				
Implements	<a href="#">osid.OsidSession</a>				
Description	<p>This session defines methods for operating on files and directories. This session is an expanded version of the DirectoryAdminSession that defines methods requiring path names for navigating a federation of directories as opposed to working within a single directory node. The directory associated with this session is the current working directory and any relative path names provided are with respect to this directory.</p>				
Method	<b>getDirectoryPath</b>				
Description	Gets the absolute path of this directory.				
Return	string	the absolute path of this directory			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>getDirectory</b>				
Description	Gets the directory associated with this session.				
Return	<a href="#">osid.filing.Directory</a>	the directory associated with this session			
Errors	<a href="#">OPERATION_FAILED</a>	unable to complete request			
	<a href="#">PERMISSION_DENIED</a>	authorization failure			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>canManageFiling</b>				
Description	Tests if this user can perform functions in this session. 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 filing management methods are not authorized, true otherwise			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>overwrite</b>				
Description	Overwrite files if a destination pathname exists.				
Parameters	boolean	overwite	true if files can be overwritten, false otherwise		
Compliance	<a href="#">mandatory</a>	This method is must be implemented.			
Method	<b>createMissingPaths</b>				
Description	Create any missing directories for a destination path that does not exist.				
Parameters	boolean	create	true if intermediate directories should be created, false otherwise		
Compliance	<a href="#">mandatory</a>	This method is must be implemented.			
Method	<b>moveFile</b>				
Description	Moves a file to another path. The destination path may be a file or directory. If the destination is a file and exists, the destination is only replaced if overwrite() is true. If the destination is a directory and does not exist, the missing directories are only created if createMissingPaths() is true.				
Parameters	string	src	the source name or path of the file		
	string	dst	the destination name or path of the directory or file		
Errors	<a href="#">ALREADY_EXISTS</a>	dst exists and overwrite() is false			
	<a href="#">INVALID_ARGUMENT</a>	src is not a file			
	<a href="#">NOT_FOUND</a>	src is not found, or the path to dst is not found and createMissingPaths() is false			
	<a href="#">NULL_ARGUMENT</a>	src or dst is null			
	<a href="#">OPERATION_FAILED</a>	unable to complete request			
	<a href="#">PERMISSION_DENIED</a>	authorization failure			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			

Method	moveDirectory				
Description	Moves a directory to another path. The destination must be a directory and if exists, the source directory is placed as a child to the given directory. If a path component in the destination does not exist, the path is created is createMissingPaths() is true.				
Parameters	string	src	the source name or path of the directory		
	string	dst	the destination name or path of the directory		
Errors	INVALID_ARGUMENT		src is not a directory		
	NOT_FOUND		src is not found, or the path to dst is not found and createMissingPaths() is false		
	NULL_ARGUMENT		src or dst is null		
	OPERATION_FAILED		unable to complete request		
	PERMISSION_DENIED		authorization failure		
Compliance	mandatory	This method must be implemented.			
Method	copyFile				
Description	Copies a file to another path. The destination path may be a file or directory. If the destination is a file and exists, the destination is only replaced if overwrite() is true. If the destination is a directory and does not exist, the missing directories are only created if createMissingPaths() is true.				
Parameters	string	src	the source name or path of the file		
	string	dst	the destination name or path of the directory or file		
Errors	ALREADY_EXISTS		dst exists and overwrite() is false		
	INVALID_ARGUMENT		src is not a file		
	NOT_FOUND		src is not found, or the path to dst is not found and createMissingPaths() is false		
	NULL_ARGUMENT		src or dst is null		
	OPERATION_FAILED		unable to complete request		
	PERMISSION_DENIED		authorization failure		
Compliance	mandatory	This method must be implemented.			
Method	copyDirectory				
Description	Copies a directory and all of its contents to another path. The destination must be a directory and if exists, the source directory is placed as a child to the given directory. If a path component in the destination does not exist, the path is created is createMissingPaths() is true.				
Parameters	string	src	the source name or path of the directory		
	string	dst	the destination name or path of the directory		
Errors	INVALID_ARGUMENT		src is not a directory		
	NOT_FOUND		src is not found, or the path to dst is not found and createMissingPaths() is false		
	NULL_ARGUMENT		src or dst is null		
	OPERATION_FAILED		unable to complete request		
	PERMISSION_DENIED		authorization failure		
Compliance	mandatory	This method must be implemented.			
Method	alias				
Description	Creates an alias from one path to another.				
Parameters	string	src	the source path		
	string	dst	the destination path		
Errors	ALREADY_EXISTS		src exists		
	NOT_FOUND		src is not found, or the path to dst is not found and createMissingPaths() is false		
	NULL_ARGUMENT		src or dst is null		
	OPERATION_FAILED		unable to complete request		
	PERMISSION_DENIED		authorization failure		
Compliance	mandatory	This method must be implemented.			

Method	unalias		
<b>Description</b>	Removes an alias.		
<b>Parameters</b>	<code>string</code>	aliasPath	the relative or absolute path to the alias.
<b>Errors</b>	<code>NOT_FOUND</code>		aliasPath not found
	<code>NULL_ARGUMENT</code>		aliasPath is null
	<code>OPERATION_FAILED</code>		unable to complete request
	<code>PERMISSION_DENIED</code>		authorization failure
<b>Compliance</b>	<code>mandatory</code>	This method must be implemented.	

<b>Interface</b>	<b>osid.filing.AllocationSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	This session defines methods for accessing space allocation and quotas.	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canAccessFilingAllocations</b>	
<b>Description</b>	Tests if this user can perform functions in this session. 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.	
<b>Return</b>	<a href="#">boolean</a>	false if filing allocation methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getAllocation</b>	
<b>Description</b>	Gets the allocation of the data store across all users. The directory available by the allocation interface may indicate a higher level directory than the one associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Allocation</a>	the allocation
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>hasQuota</b>	
<b>Description</b>	Tests if there is a quota associated with this directory.	
<b>Return</b>	<a href="#">boolean</a>	true if a quota exists, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getQuotaAllocation</b>	
<b>Description</b>	Gets the quota allocation of the data store for this user. The directory available by the allocation interface may indicate a higher level directory than the one requested if the one requested is a sub-directory inside the qoutad data store.	
<b>Return</b>	<a href="#">osid.filing.Allocation</a>	the allocation
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a>	hasQuota() is false
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.AllocationAdminSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	This session defines methods for managing quotas.	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canAccessAllocations</b>	
<b>Description</b>	Tests if this user can perform lookup functions in this session. 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.	
<b>Return</b>	<a href="#">boolean</a>	false if filing allocation lookup methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>hasQuota</b>	
<b>Description</b>	Tests if the given user has a quota in this directory.	
<b>Parameters</b>	<a href="#">osid.id.Id</a>	agentId
<b>Return</b>	<a href="#">boolean</a>	true if a quota exists, false otherwise
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	agentId is null
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getQuotaAllocation</b>	
<b>Description</b>	Gets the quota allocation of the data store for a given user. The directory available by the allocation interface may indicate a higher level directory than the one requested if the one requested is a sub-directory inside the qoutad data store.	
<b>Parameters</b>	<a href="#">osid.id.Id</a>	agentId
<b>Return</b>	<a href="#">osid.filing.Allocation</a>	the allocation
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a>	hasQuota(agentId) is false
	<a href="#">NULL_ARGUMENT</a>	agentId is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canSetAllocations</b>	
<b>Description</b>	Tests if this user can perform quota assignments in this session. 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.	
<b>Return</b>	<a href="#">boolean</a>	false if filing allocation assignment methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	setQuotaSpace				
Description	Sets the quota allocation for a given user for this directory.				
Parameters	osid.id.Id	agentId	the agent Id of the user		
	cardinal	size	space in bytes		
Errors	NOT_FOUND		agentId is not found		
	NULL_ARGUMENT		agentId is null		
	OPERATION_FAILED		unable to complete request		
	PERMISSION_DENIED		authorization failure		
Compliance	mandatory	This method must be implemented.			
Method	setQuotaFiles				
Description	Sets the quota allocation for a given user for this directory.				
Parameters	osid.id.Id	agentId	the agent Id of the user		
	cardinal	numFiles	number of files		
Errors	NOT_FOUND		agentId is not found		
	NULL_ARGUMENT		agentId is null		
	OPERATION_FAILED		unable to complete request		
	PERMISSION_DENIED		authorization failure		
Compliance	mandatory	This method must be implemented.			

<b>Interface</b>	<b>osid.filing.AllocationNotificationSession</b>	
<b>Implements</b>	<a href="#">osid.OsidSession</a>	
<b>Description</b>	This session defines methods to receive notifications of allocation warnings and changed quota assignments. 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.	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path of this directory.	
<b>Return</b>	<a href="#">string</a>	the absolute path of this directory
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the Directory associated with this session.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the Directory associated with this session
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>canRegisterForAllocationNotifications</b>	
<b>Description</b>	Tests if this user can register for Directory 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.	
<b>Return</b>	<a href="#">boolean</a>	false if notification methods are not authorized, true otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>registerForAllocationWarnings</b>	
<b>Description</b>	Register for notifications of allocation warnings. AllocationReceiver.warning() is invoked when a new warning is issued.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>registerForAllocationClearWarnings</b>	
<b>Description</b>	Register for notifications of clearing of allocation warnings. AllocationReceiver.clearWarning() is invoked when a warning is cleared.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>registerForChangedQuota</b>	
<b>Description</b>	Registers for notification of new, updated or changed quotas. AllocationReceiver.changedQuota() is invoked when a quota is changed.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryEntry</b>	
<b>Implements</b>		
<b>Description</b>	DirectoryEntry defines methods in common to both File and Directory.	
<b>Method</b>	<b>getName</b>	
<b>Description</b>	Gets the name of this entry. The name does not include the path. If this entry represents an alias, the name of the alias is returned.	
<b>Return</b>	<a href="#">string</a>	the entry name
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getAlias</b>	
<b>Description</b>	Tests if this entry is an alias.	
<b>Return</b>	<a href="#">boolean</a>	true if this is an alias, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getPath</b>	
<b>Description</b>	Gets the full path of this entry. The path includes the name. Path components are separated by a /. If this entry represents an alias, the path to the alias is returned.	
<b>Return</b>	<a href="#">string</a>	the path
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getRealPath</b>	
<b>Description</b>	Gets the real path of this entry. The path includes the name. Path components are separated by a /. If this entry represents an alias, the full path to the target file or directory is returned.	
<b>Return</b>	<a href="#">string</a>	the path
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getOwnerId</b>	
<b>Description</b>	Gets the Id of the Agent that owns this entry.	
<b>Return</b>	<a href="#">osid.id.Id</a>	the Agent Id
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getOwner</b>	
<b>Description</b>	Gets the Agent that owns this entry.	
<b>Return</b>	<a href="#">osid.authentication.Agent</a>	the Agent
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	authentication service not available
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getCreatedTime</b>	
<b>Description</b>	Gets the created time of this entry.	
<b>Return</b>	<a href="#">osid.calendaring.DateTime</a>	the created time
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getLastModifiedTime</b>	
<b>Description</b>	Gets the last modified time of this entry.	
<b>Return</b>	<a href="#">osid.calendaring.DateTime</a>	the last modified time
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getLastAccessTime</b>	
<b>Description</b>	Gets the last access time of this entry.	
<b>Return</b>	<a href="#">osid.calendaring.DateTime</a>	the last access time
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getRecordTypes</b>	
<b>Description</b>	Gets the record types of this entry. The Type explicitly indicates the specification of the record interface.	
<b>Return</b>	<a href="#">osid.type.TypeList</a>	the record types
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	hasRecordType		
Description	Tests if this entry supports the given record Type. The given type may be supported through inheritance.		
Parameters	osid.type.Type	recordType	a record type
Return	boolean		
Errors	NULL_ARGUMENT		
Compliance	mandatory		
Method	getGenusType		
Description	Gets the genus type of this object.		
Return	osid.type.Type		the genus type of this object
Compliance	mandatory		
Method	isOfGenusType		
Description	Tests if this object is of the given genus Type. The given genus type may be supported by the object through the type hierarchy.		
Parameters	osid.type.Type	genusType	a genus type
Return	boolean		
Errors	NULL_ARGUMENT		
Compliance	mandatory		
	isCurrent		
Description	Tests to see if the last method invoked retrieved up-to-date data. Simple retrieval methods do not specify errors as, generally, the data is retrieved once at the time this object is instantiated. Some implementations may provide real-time data though the application may not always care. An implementation providing a real-time service may fall back to a previous snapshot in case of error. This method returns false if the data last retrieved was stale.		
Return	boolean		true if the last data retrieval was up to date, false otherwise
Compliance	mandatory		
Provider Notes	Providers should return false unless all getters are implemented using real-time queries, or some trigger process keeps the data in this object current. Providers should populate basic data elements at the time this object is instantiated, or set an error, to ensure some data availability.		
Method	getProperties		
Description	Gets a list of properties corresponding to the interface type of this object. Properties provide a means for applications to display a representation of the contents of an object without understanding its Type specification. Applications needing to examine a specific property or perform updates should use the methods defined by the object's interface Type.		
Return	osid.PropertyList		a list of properties
Errors	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		an authorization failure occurred
Compliance	mandatory		

Method	getPropertiesByRecordType		
<b>Description</b>	Gets a list of properties corresponding to the specified record type. Properties provide a means for applications to display a representation of the contents of an object without understanding its record type specification. Applications needing to examine a specific property or perform updates should use the methods defined by the object interface Type. The resulting set includes properties specified by parents of the record type in the type hierarchy.		
<b>Parameters</b>	<a href="#">osid.type.Type</a>	recordType	the record type corresponding to the properties set to retrieve
<b>Return</b>	<a href="#">osid.PropertyList</a>		
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>		recordType is null
	<a href="#">OPERATION_FAILED</a>		unable to complete request
	<a href="#">PERMISSION_DENIED</a>		an authorization failure occurred
	<a href="#">UNSUPPORTED</a>		hasRecordType(recordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>		This method must be implemented.

Interface	osid.filing.DirectoryEntryQuery				
Implements					
Description	DirectoryEntryQuery defines methods in common to both FileQuery and DirectoryQuery.				
Method	<b>getStringMatchTypes</b>				
Description	Gets the string matching types supported.				
Return	<a href="#">osid.type.TypeList</a>	a list containing the supported string match types			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>supportsStringMatchType</b>				
Description	Tests if the given string matching type is supported.				
Parameters	<a href="#">osid.type.Type</a>	searchType	a Type indicating a string match type		
Return	<a href="#">boolean</a>	true if the given Type is supported, false otherwise			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>matchName</b>				
Description	Matches entry names. Supplying multiple strings behaves like a boolean AND among the elements each which must correspond to the stringMatchType. An OR can be performed with multiple query interfaces.				
Parameters	<a href="#">string</a>	name	name to match		
	<a href="#">osid.type.Type</a>	stringMatchType	the string match type		
	<a href="#">boolean</a>	match	true for a positive match, false for a negative match		
Errors	<a href="#">INVALID_ARGUMENT</a>		name not of stringMatchType		
	<a href="#">NULL_ARGUMENT</a>		name or stringMatchType is null		
	<a href="#">UNSUPPORTED</a>		supportsStringMatchType(stringMatchType) is false		
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>matchPath</b>				
Description	Matches an absolute pathname of a directory entry. Supplying multiple strings behaves like a boolean AND among the elements each which must correspond to the stringMatchType. An OR can be performed with multiple query interfaces.				
Parameters	<a href="#">string</a>	path	path to match		
	<a href="#">osid.type.Type</a>	stringMatchType	the string match type		
	<a href="#">boolean</a>	match	true for a positive match, false for a negative match		
Errors	<a href="#">INVALID_ARGUMENT</a>		name not of stringMatchType		
	<a href="#">NULL_ARGUMENT</a>		path or stringMatchType is null		
	<a href="#">UNSUPPORTED</a>		supportsStringMatchType(stringMatchType) is false		
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>supportsDirectoryQuery</b>				
Description	Tests if a DirectoryQuery is available.				
Return	<a href="#">boolean</a>	true if a directory query interface is available, false otherwise			
Compliance	<a href="#">mandatory</a>	This method must be implemented.			
Method	<b>getDirectoryQuery</b>				
Description	Gets the query interface for a directory to match the parent directory. There is only one DirectoryQuery per DirectoryEntryQuery. Multiple retrievals return the same object.				
Return	<a href="#">osid.filing.DirectoryQuery</a>	the directory query			
Errors	<a href="#">UNIMPLEMENTED</a>		supportsDirectoryQuery() is false		
Compliance	<a href="#">optional</a>	This method must be implemented if supportsDirectoryQuery() is true.			

<b>Method</b>	<b>matchAliases</b>	
<b>Description</b>	Matches aliases only.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>matchTargets</b>	
<b>Description</b>	Matches target files only, omitting aliases.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>matchAll</b>	
<b>Description</b>	Matches both targets and aliases.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>matchOwnerId</b>	
<b>Description</b>	Matches files whose entries are owned by the given agent id.	
<b>Parameters</b>	<code>osid.id.Id</code>	agentId
	<code>boolean</code>	match
<b>Errors</b>	<code>NULL_ARGUMENT</code>	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>supportsAgentQuery</b>	
<b>Description</b>	Tests if an AgentQuery is available for querying agents.	
<b>Return</b>	<code>boolean</code>	true if an agent query interface is available, false otherwise
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>getAgentQuery</b>	
<b>Description</b>	Gets the query interface for an agent.	
<b>Return</b>	<code>osid.authentication.AgentQuery</code>	the agent query
<b>Errors</b>	<code>UNIMPLEMENTED</code>	
<b>Compliance</b>	<b>optional</b>	This method must be implemented if supportsAgentQuery() is true.
<b>Method</b>	<b>matchCreatedTime</b>	
<b>Description</b>	Match directory entries that are created between the specified time period.	
<b>Parameters</b>	<code>osid.calendaring.DateTime</code>	start
	<code>osid.calendaring.DateTime</code>	end
	<code>boolean</code>	match
<b>Errors</b>	<code>INVALID_ARGUMENT</code>	
	<code>NULL_ARGUMENT</code>	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>matchModifiedTime</b>	
<b>Description</b>	Match directory entries that are modified between the specified time period.	
<b>Parameters</b>	<code>osid.calendaring.DateTime</code>	start
	<code>osid.calendaring.DateTime</code>	end
	<code>boolean</code>	match
<b>Errors</b>	<code>INVALID_ARGUMENT</code>	
	<code>NULL_ARGUMENT</code>	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

Method	matchLastAccessTime		
Description	Match directory entries that were last accessed between the specified time period.		
Parameters	osid.calendaring.DateTime	start	start time of the query
	osid.calendaring.DateTime	end	end time of the query
	boolean	match	true for a positive match, false for a negative match
Errors	INVALID_ARGUMENT		end is less than start
	NULL_ARGUMENT		start or end is null
Compliance	mandatory		This method must be implemented.
Method	matchGenusType		
Description	Sets a Type for querying objects of a given genus.		
Parameters	osid.type.Type	genusType	the object genus type
	boolean	match	true for a positive match, false for a negative match
Errors	NULL_ARGUMENT		genusType is null
Compliance	mandatory		This method must be implemented.
Method	matchRecordType		
Description	Sets a Type for querying objects of a given record type.		
Parameters	osid.type.Type	recordType	the entry record type
	boolean	match	true for a positive match, false for a negative match
Errors	NULL_ARGUMENT		recordType is null
Compliance	mandatory		This method must be implemented.
Method	hasRecordType		
Description	Tests if this query supports the given record Type. The given record type may be supported by the object through interface/type inheritance. This method should be checked before retrieving the record interface.		
Parameters	osid.type.Type	recordType	a type
Return	boolean		true if a record query of the given record Type is available, false otherwise
Errors	NULL_ARGUMENT		recordType is null
Compliance	mandatory		This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryEntryForm</b>	
<b>Implements</b>		
<b>Description</b>	DirectoryEntryForm defines methods in common to both FileForm and DirectoryForm.	
<b>Method</b>	<b>getCommentMetadata</b>	
<b>Description</b>	Gets the metadata for the comment corresponding to this form submission. The comment is used for describing the nature of the change to the corresponding object for the purposes of logging and auditing.	
<b>Return</b>	<a href="#">osid.Metadata</a>	metadata for the comment
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>setComment</b>	
<b>Description</b>	Sets a comment.	
<b>Parameters</b>	<a href="#">string</a>	comment
	<a href="#">INVALID_ARGUMENT</a>	comment is invalid
<b>Errors</b>	<a href="#">NO_ACCESS</a>	comment cannot be modified
	<a href="#">NULL_ARGUMENT</a>	comment is null
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>validate</b>	
<b>Description</b>	Performs a validation of the data set in this form. This method returns nothing if validation is not performed or validation is successful. An transaction is returned if the form is invalid.	
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>	attempt to perform validation failed
	<a href="#">TRANSACTION_FAILURE</a>	form is invalid
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getNameMetadata</b>	
<b>Description</b>	Gets the metadata for the name.	
<b>Return</b>	<a href="#">osid.Metadata</a>	metadata for the name
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>setName</b>	
<b>Description</b>	Sets the name.	
<b>Parameters</b>	<a href="#">string</a>	name
	<a href="#">INVALID_ARGUMENT</a>	name is invalid
<b>Errors</b>	<a href="#">NO_ACCESS</a>	name cannot be modified
	<a href="#">NULL_ARGUMENT</a>	displayName is null
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getOwnerMetadata</b>	
<b>Description</b>	Gets the metadata for the owner.	
<b>Return</b>	<a href="#">osid.Metadata</a>	metadata for the owner
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>setOwner</b>	
<b>Description</b>	Sets the owner.	
<b>Parameters</b>	<a href="#">osid.id.Id</a>	agentId
	<a href="#">INVALID_ARGUMENT</a>	agentId is invalid
<b>Errors</b>	<a href="#">NO_ACCESS</a>	owner cannot be modified
	<a href="#">NULL_ARGUMENT</a>	agentId is null
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getGenusMetadata</b>	
<b>Description</b>	Gets the metadata for the genus type.	
<b>Return</b>	<a href="#">osid.Metadata</a>	metadata for the genus
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Method	setGenus		
<b>Description</b>	Sets the genus type.		
<b>Parameters</b>	osid.type.Type	genusType	the new genus type
<b>Errors</b>	INVALID_ARGUMENT		genusType is invalid
	NO_ACCESS		genus cannot be modified
	NULL_ARGUMENT		genusType is null
<b>Compliance</b>	mandatory		This method must be implemented.
Method	hasRecordType		
<b>Description</b>	Tests if this form supports the given record Type. The given record type may be supported by the object through interface/type inheritance. This method should be checked before retrieving the record interface.		
<b>Parameters</b>	osid.type.Type	recordType	a record type
<b>Return</b>	boolean		true if a record form of the given record Type is available, false otherwise
<b>Errors</b>	NULL_ARGUMENT		recordType is null
<b>Compliance</b>	mandatory		This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryEntrySearchOrder</b>	
<b>Implements</b>		
<b>Description</b>	An interface for specifying the order of search results.	
<b>Method</b>	<b>ascend</b>	
<b>Description</b>	Specifies a preference for ordering the result set in an ascending manner.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>descend</b>	
<b>Description</b>	Specifies a preference for ordering the result set in a descending manner.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByName</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry name.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByPath</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry path.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByOwner</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry owner.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByCreatedTime</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry creation time.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByModifiedTime</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry modification time.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByLastAccessTime</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry last access time.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>orderByGenusType</b>	
<b>Description</b>	Specifies a preference for ordering the result set by the entry genus type.	
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.
<b>Method</b>	<b>hasRecordType</b>	
<b>Description</b>	Tests if this search order supports the given record Type. The given record type may be supported by the object through interface/type inheritance. This method should be checked before retrieving the record interface.	
<b>Parameters</b>	<b>osid.type.Type</b>	recordType a type
<b>Return</b>	<b>boolean</b>	true if an order record of the given record Type is available, false otherwise
<b>Errors</b>	<b>NULL_ARGUMENT</b>	recordType is null
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.File</b>	
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntry</a>	
<b>Description</b>	File represents a file in a file system.	
<b>Method</b>	<b>hasSize</b>	
<b>Description</b>	Tests if this file has a known size.	
<b>Return</b>	<a href="#">boolean</a>	true if this file has a size, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getSize</b>	
<b>Description</b>	Gets the size of this file in bytes if hasSize() is true.	
<b>Return</b>	<a href="#">cardinal</a>	the size of this file
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a>	hasSize() is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
	<b>getFileRecord</b>	
<b>Description</b>	Gets the record corresponding to the given File record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces. The fileRecordType may be the Type returned in getRecordTypes() or any of its parents in a Type hierarchy where hasRecordType(fileRecordType) is true.	
<b>Parameters</b>	<a href="#">osid.type.Type</a> fileInterfaceType	the file record type
<b>Return</b>	<a href="#">osid.filing.File</a>	the file record
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a> <a href="#">OPERATION_FAILED</a> <a href="#">PERMISSION_DENIED</a> <a href="#">UNSUPPORTED</a>	fileRecordType is null unable to complete request authorization failure occurred hasInterfaceType(fileRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a File. The methods specified by the record type are available through the underlying object.	
<b>getFile</b>		
<b>Description</b>	Gets the File from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.File</a>	the file
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

Interface	osid.filing.FileQuery		
Implements	<a href="#">osid.filing.DirectoryEntryQuery</a>		
Description	This is the query interface for searching files. Each method specifies an AND term while multiple invocations of the same method produces a nested OR.		
Method	matchSize		
Description	Matches files whose size is within and including the given range.		
Parameters	cardinal	from	low file size
	cardinal	to	high file size
	boolean	match	true for a positive match, false for a negative match
Errors	<a href="#">INVALID_ARGUMENT</a>		to is less than from
Compliance	<a href="#">mandatory</a>		This method must be implemented.
Method	matchDataString		
Description	Adds data strings to this query to match files whose content contains these strings. Supplying multiple strings behaves like a boolean AND among the elements each which must correspond to the stringMatchType. An OR can be performed with multiple query interfaces.		
Parameters	string	data	string to match
	<a href="#">osid.type.Type</a>	stringMatchType	the string match type
	boolean	match	true for a positive match, false for a negative match
Errors	<a href="#">INVALID_ARGUMENT</a>		data not of stringMatchType
	<a href="#">NULL_ARGUMENT</a>		data or stringMatchType is null
	<a href="#">UNSUPPORTED</a>		supportsStringMatchType(stringMatchType) is false
Compliance	<a href="#">mandatory</a>		This method must be implemented.
Method	matchData		
Description	Matches files who data contains the given bytes.		
Parameters	byte[]	data	data to match
	boolean	match	true for a positive match, false for a negative match
Errors	<a href="#">NULL_ARGUMENT</a>		data is null
Compliance	<a href="#">mandatory</a>		This method must be implemented.
Method	getFileQueryRecord		
Description	Gets the record query interface corresponding to the given File record Type. Multiple record retrievals produce a nested boolean OR term.		
Parameters	<a href="#">osid.type.Type</a>	fileRecordType	a file record type
Return	<a href="#">osid.filing.FileQuery</a>		the file query record
Errors	<a href="#">NULL_ARGUMENT</a>		fileRecordType is null
	<a href="#">OPERATION_FAILED</a>		unable to complete request
	<a href="#">PERMISSION_DENIED</a>		authorization failure occurred
	<a href="#">UNSUPPORTED</a>		hasRecordType(fileRecordType) is false
Compliance	<a href="#">mandatory</a>		This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileQueryRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a FileQuery. The methods specified by the record type are available through the underlying object.	
<b>getFileQuery</b>		
<b>Description</b>	Gets the FileQuery from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.FileQuery</a>	the file query
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileForm</b>	
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntryForm</a>	
<b>Description</b>	This is the form for creating and updating Files. Like all OsidForm objects, various data elements may be set here for use in the create and update methods in the FileAdminSession. For each data element that may be set, metadata may be examined to provide display hints or data constraints.	
<b>Method</b>	<b>getFileFormRecord</b>	
<b>Description</b>	Gets the FileFormRecord interface corresponding to the given log record interface Type.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileRecordType the file record type
<b>Return</b>	<a href="#">osid.filing.FileFormRecord</a>	the record
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasRecordType(fileRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileFormRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a FileForm. The methods specified by the record type are available through the underlying object.	
<b>getFileForm</b>		
<b>Description</b>	Gets the FileForm from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.FileForm</a>	the file form
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileSearchOrder</b>	
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntrySearchOrder</a>	
<b>Description</b>	An interface for specifying the ordering of search results.	
<b>Method</b>	<b>orderBySize</b>	
<b>Description</b>	Specifies a preference for ordering the results by file size.	
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getFileSearchOrderRecord</b>	
<b>Description</b>	Gets the file search record order corresponding to the given file record Type. Multiple retrievals return the same underlying object.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileRecordType a file record type
<b>Return</b>	<a href="#">osid.filing.FileSearchOrderRecord</a>	the file search order record interface
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasRecordType(fileRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileSearchOrderRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a FileSearchOrder. The methods specified by the record type are available through the underlying object.	
<b>getFileSearchOrder</b>		
<b>Description</b>	Gets the FileSearchOrder from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.FileSearchOrder</a>	the file search order
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileSearch</b>	
<b>Implements</b>	<a href="#">osid.OsidSearch</a>	
<b>Description</b>	The search interface for governing file searches.	
<b>Method</b>	<b>searchWithinFileResults</b>	
<b>Description</b>	Execute this search using a previous search result.	
<b>Parameters</b>	<a href="#">osid.filing.FileSearchResults</a>	results results from a query
<b>Errors</b>	<a href="#">INVALID_ARGUMENT</a>	results is not valid
	<a href="#">NULL_ARGUMENT</a>	results is null
<b>Compliance</b>	mandatory This method must be implemented.	
<b>Method</b>	<b>searchAmongFiles</b>	
<b>Description</b>	Execute this search among the given list of files.	
<b>Parameters</b>	<a href="#">string[]</a>	files list of files
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	files is null
<b>Compliance</b>	mandatory This method must be implemented.	
<b>Method</b>	<b>orderFileResults</b>	
<b>Description</b>	Specify an ordering to the search results.	
<b>Parameters</b>	<a href="#">osid.filing.FileSearchOrder</a>	fileSearchOrder file search order
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileSearchOrder is null
	<a href="#">UNSUPPORTED</a>	fileSearchOrder is not of this service
<b>Compliance</b>	mandatory This method must be implemented.	
<b>Method</b>	<b>getFileSearchRecord</b>	
<b>Description</b>	Gets the record corresponding to the given file search record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileSearchRecordType a file search record type
<b>Return</b>	<a href="#">osid.filing.FileSearchRecord</a>	the file search interface
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileSearchRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasSearchRecordType(fileSearchRecordType) is false
<b>Compliance</b>	mandatory This method must be implemented.	

<b>Interface</b>	<b>osid.filing.FileSearchRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a FileSearch. The methods specified by the record type are available through the underlying object.	
<b>getFileSearch</b>		
<b>Description</b>	Gets the FileSearch from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.FileSearch</a>	the file search
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileSearchResults</b>	
<b>Implements</b>	<a href="#">osid.OsidSearchResults</a>	
<b>Description</b>	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.	
<b>Method</b>	<b>getFiles</b>	
<b>Description</b>	Gets the file list resulting from a search.	
<b>Return</b>	<a href="#">osid.filing.FileList</a>	the directory list
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getFileSearchResultsRecord</b>	
<b>Description</b>	Gets the record corresponding to the given file search record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileSearchRecordType a file search record type
<b>Return</b>	<a href="#">osid.filing.FileSearchResultsRecord</a>	the file search interface
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	fileSearchRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasSearchRecordType(fileSearchRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileSearchResultsRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a FileSearchResults. The methods specified by the record type are available through the underlying object.	
<b>getFileSearchResults</b>		
<b>Description</b>	Gets the FileSearchResults from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.FileSearchResults</a>	the file search results
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.FileReceiver</b>				
<b>Implements</b>	<a href="#">osid.OsidReceiver</a>				
<b>Description</b>	The file receiver is the consumer supplied interface for receiving notifications pertaining to new, updated or deleted files.				
<b>Method</b>	<b>newFile</b>				
<b>Description</b>	The callback for notifications of new files.				
<b>Parameters</b>	<a href="#">string</a>	filePath	absolute pathname to the new file		
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.			
<b>Method</b>	<b>changedFile</b>				
<b>Description</b>	The callback for notification of updated files.				
<b>Parameters</b>	<a href="#">string</a>	filePath	absolute pathname to the changed file		
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.			
<b>Method</b>	<b>deletedFile</b>				
<b>Description</b>	the callback for notification of deleted files.				
<b>Parameters</b>	<a href="#">string</a>	filePath	absolute pathname to the deleted file		
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.			

<b>Interface</b>	<b>osid.filing.FileList</b>	
<b>Implements</b>	<a href="#">osid.OsidList</a>	
<b>Description</b>	<p>Like all OsidLists, FileList provides a means for accessing File elements sequentially either one at a time or many at a time. Examples:</p> <pre>while (el.hasNext()) {     Entry entry = el.getNextEntry(); }</pre> <p>or</p> <pre>while (el.hasNext()) {     Entry[] entries = e</pre>	
<b>Method</b>	<b>getNextFile</b>	
<b>Description</b>	Gets the next File in this list.	
<b>Return</b>	<a href="#">osid.filing.File</a>	the next File in this list. The hasNext() method should be used to test that a next File is available before calling this method.
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a> <a href="#">OPERATION_FAILED</a>	no more elements available in this list unable to complete request
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getNextFiles</b>	
<b>Description</b>	Gets the next set of File elements in this list which must be less than or equal to the number returned from available().	
<b>Parameters</b>	<a href="#">cardinal</a>	<a href="#">n</a>
		the number of File elements requested which should be less than or equal to available()
<b>Return</b>	<a href="#">osid.filing.File[]</a>	an array of File elements. The length of the array is less than or equal to the number specified.
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a> <a href="#">OPERATION_FAILED</a>	no more elements available in this list unable to complete request
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.Directory</b>	
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntry</a>	
<b>Description</b>	Directory represents a directory in a file system that may contain other files and directories.	
	<b>getDirectoryRecord</b>	
<b>Description</b>	Gets the record corresponding to the given Directory record Type. This method must be used to retrieve an object implementing the requested record interface along with all of its ancestor interfaces. The directoryRecordType may be the Type returned in getRecordTypes() or any of its parents in a Type hierarchy where hasRecordType(directoryRecordType) is true.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	directoryRecordType the directory record type
<b>Return</b>	<a href="#">osid.filing.Directory</a>	
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	directoryRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasInterfaceType(directoryRecordType) is false
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a Directory. The methods specified by the record type are available through the underlying object.	
<b>getDirectory</b>		
<b>Description</b>	Gets the Directory from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the directory
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryQuery</b>		
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntryQuery</a>		
<b>Description</b>	<p>This is the query interface for searching directories. Each method match request produces an AND term while multiple invocations of a method produces a nested OR, except for accessing the DirectoryQuery subinterface.</p> <p>An example to find directories whose name is "Library".</p> <pre>DirectoryQuery query = session.getDirectoryQuery(); query.matchName("Library", wordStringMatchType, true);  DirectoryList list = session.getDirectoriesByQuery(query);</pre>		
<b>Method</b>	<b>supportsFileQuery</b>		
<b>Description</b>	Tests if a FileQuery is available.		
<b>Return</b>	<a href="#">boolean</a>		true if a file query interface is available, false otherwise
<b>Compliance</b>	<a href="#">mandatory</a>		This method must be implemented.
<b>Method</b>	<b>getFileQuery</b>		
<b>Description</b>	Gets the query interface for a file contained within the directory.		
<b>Parameters</b>	<a href="#">boolean</a>	match	true for a positive match, false for a negative match
<b>Return</b>	<a href="#">osid.filing.FileQuery</a>		the directory query
<b>Errors</b>	<a href="#">UNIMPLEMENTED</a>		supportsFileQuery() is false
<b>Compliance</b>	<a href="#">optional</a>		This method must be implemented if supportsFileQuery() is true.
<b>Method</b>	<b>getFileQueryForInterfaceType</b>		
<b>Description</b>	Gets the file query interface for the file type. Supported types are defined in the FilingManager.		
<b>Parameters</b>	<a href="#">osid.type.Type</a>	fileInterfaceType	a file interface type
	<a href="#">boolean</a>	match	true for a positive match, false for negative match
<b>Return</b>	<a href="#">osid.filing.DirectoryQuery</a>		the file query
	<a href="#">NULL_ARGUMENT</a>		fileInterfaceType is null
<b>Errors</b>	<a href="#">UNIMPLEMENTED</a>		supportFileQuery() is false
	<a href="#">UNSUPPORTED</a>		FilingManager.supportsFileInterfaceType(fileInterfaceType) is false
<b>Compliance</b>	<a href="#">optional</a>		This method must be implemented if supportsFileQuery() is true.
<b>Method</b>	<b>getDirectoryQueryRecord</b>		
<b>Description</b>	Gets the record query interface corresponding to the given Directory record Type. Multiple record retrievals produce a nested boolean OR term.		
<b>Parameters</b>	<a href="#">osid.type.Type</a>	directoryRecordType	a directory record type
<b>Return</b>	<a href="#">osid.filing.DirectoryQuery</a>		the directory query record
	<a href="#">NULL_ARGUMENT</a>		directoryRecordType is null
<b>Errors</b>	<a href="#">OPERATION_FAILED</a>		unable to complete request
	<a href="#">PERMISSION_DENIED</a>		authorization failure occurred
	<a href="#">UNSUPPORTED</a>		hasRecordType(directoryRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>		This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryQueryRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a DirectoryQuery. The methods specified by the record type are available through the underlying object.	
<b>getDirectoryQuery</b>		
<b>Description</b>	Gets the DirectoryQuery from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.DirectoryQuery</a>	the directory query
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryForm</b>	
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntryForm</a>	
<b>Description</b>	This is the form for creating and updating Directory objects. Like all OsidForm objects, various data elements may be set here for use in the create and update methods in the DirectoryAdminSession. For each data element that may be set, metadata may be examined to provide display hints or data constraints.	
<b>Method</b>	<b>getDirectoryFormRecord</b>	
<b>Description</b>	Gets the DirectoryFormRecord interface corresponding to the given log record interface Type.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	directoryRecordType the directory record type
<b>Return</b>	<a href="#">osid.filing.DirectoryFormRecord</a>	the record
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	directoryRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasRecordType(directoryRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryFormRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a DirectoryForm. The methods specified by the record type are available through the underlying object.	
<b>getDirectoryForm</b>		
<b>Description</b>	Gets the DirectoryForm from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.DirectoryForm</a>	the directory form
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectorySearchOrder</b>	
<b>Implements</b>	<a href="#">osid.filing.DirectoryEntrySearchOrder</a>	
<b>Description</b>	An interface for specifying the ordering of search results.	
<b>Method</b>	<b>getFileSearchOrderRecord</b>	
<b>Description</b>	Gets the directory search record order corresponding to the given directory record Type. Multiple retrievals return the same underlying object.	
<b>Parameters</b>	<a href="#">osid.type.Type</a>	directoryRecordType a directory record type
<b>Return</b>	<a href="#">osid.filing.DirectorySearchOrderRecord</a>	
<b>Errors</b>	<a href="#">NULL_ARGUMENT</a>	directoryRecordType is null
	<a href="#">OPERATION_FAILED</a>	unable to complete request
	<a href="#">PERMISSION_DENIED</a>	authorization failure occurred
	<a href="#">UNSUPPORTED</a>	hasRecordType(directoryRecordType) is false
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectorySearchOrderRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a DirectorySearchOrder. The methods specified by the record type are available through the underlying object.	
<b>getDirectorySearchOrder</b>		
<b>Description</b>	Gets the DirectorySearchOrder from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.DirectorySearchOrder</a>	the directory search order
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

Interface	osid.filing.DirectorySearch	
Implements	<a href="#">osid.OsidSearch</a>	
Description	<p>The search interface for governing directory searches.</p> <p>An example to find directories whose name is "Library" sorted by the path name.</p> <pre>DirectoryQuery query = session.getDirectoryQuery(); query.addNameMatch("Library", wordStringMatchType, true);  DirectorySearch search = session.getDirectorySearch(); search.orderByPath();  DirectorySearchResults results = session.getDirectoriesBySearch(query, search); DirectoryList directories = results.getDirectories();</pre>	
Method	<b>searchWithinDirectoryResults</b>	
Description	Execute this search using a previous search result.	
Parameters	<a href="#">osid.filing.DirectorySearchResults</a>	results
Errors	<a href="#">INVALID_ARGUMENT</a> <a href="#">NULL_ARGUMENT</a>	
Compliance	mandatory	
Method	<b>searchAmongDirectories</b>	
Description	Execute this search among the given list of directories.	
Parameters	<a href="#">string[]</a>	directories
Errors	<a href="#">NULL_ARGUMENT</a> <a href="#">UNSUPPORTED</a>	
Compliance	mandatory	
Method	<b>orderDirectoryResults</b>	
Description	Specify an ordering to the search results.	
Parameters	<a href="#">osid.filing.DirectorySearchOrder</a>	directorySearchOrder
Errors	<a href="#">NULL_ARGUMENT</a> <a href="#">UNSUPPORTED</a>	
Compliance	mandatory	
Method	<b>getDirectorySearchRecord</b>	
Description	Gets the record corresponding to the given directory 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	<a href="#">osid.type.Type</a>	directorySearchRecordType
Return	<a href="#">osid.filing.DirectorySearchRecord</a>	
Errors	<a href="#">NULL_ARGUMENT</a> <a href="#">OPERATION_FAILED</a> <a href="#">PERMISSION_DENIED</a> <a href="#">UNSUPPORTED</a>	
Compliance	mandatory	

<b>Interface</b>	<b>osid.filing.DirectorySearchRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a DirectorySearch. The methods specified by the record type are available through the underlying object.	
<b>getDirectorySearch</b>		
<b>Description</b>	Gets the DirectorySearch from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.DirectorySearch</a>	the directory search
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

Interface	osid.filing.DirectorySearchResults	
Implements	<a href="#">osid.OsidSearchResults</a>	
Description	<p>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. This example gets a result set from a keyword match.</p> <p>An example to find directories whose path contains "System" from a resulting search of directories whose name is "Library", sorted by the path name.</p> <pre>DirectoryQuery query = session.getDirectoryQuery(); query.addNameMatch("Library", wordStringMatchType, true); DirectorySearch search = session.getDirectorySearch(); DirectorySearchResults results = session.getDirectoriesBySearch(query, search);  query = session.getDirectoryQuery(); query.addNameMatch("System", wordStringMatchType, true); search = session.getDirectorySearch(); search.searchWithinDirectoryResults(results); DirectorySearchOrder order = session.getDirectorySearchOrder(); order.orderByPath(); search.orderDirectoryResults(order);  results = session.getDirectoriesBySearch(query, search); DirectoryList directories = results.getDirectories();</pre>	
Method	<b>getDirectories</b>	
Description	Gets the directory list resulting from a search.	
Return	<a href="#">osid.filing.DirectoryList</a>	the directory list
Compliance	<a href="#">mandatory</a>	This method must be implemented.
Method	<b>getDirectorySearchResultsRecord</b>	
Description	Gets the record corresponding to the given directory 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	<a href="#">osid.type.Type</a> directorySearchRecordType	a directory search record type
Return	<a href="#">osid.filing.DirectorySearchResultsRecord</a>	the directory search interface
Errors	NULL_ARGUMENT	directorySearchRecordType is null
	OPERATION_FAILED	unable to complete request
	PERMISSION_DENIED	authorization failure occurred
	UNSUPPORTED	hasSearchRecordType(directorySearchRecordType) is false
Compliance	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectorySearchResultsRecord</b>	
<b>Implements</b>	<a href="#">osid.OsidRecord</a>	
<b>Description</b>	A record for a DirectorySearchResults. The methods specified by the record type are available through the underlying object.	
<b>getDirectorySearchResults</b>		
<b>Description</b>	Gets the DirectorySearchResults from which this record originated.	
<b>Return</b>	<a href="#">osid.filing.DirectorySearchResults</a>	the directory search results
<b>Compliance</b>	<b>mandatory</b>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryReceiver</b>
<b>Implements</b>	<a href="#">osid.OsidReceiver</a>
<b>Description</b>	<p>The directory receiver is the consumer supplied interface for receiving notifications pertaining to new, updated or deleted directories.</p> <p>An example showing how to receive notifications of new directories created under a user's home directory.</p> <pre>class myReceiver {     void newDirectory(string path) {         print "new directory created: " + path;     }      void changedDirectory(string path) {}     void deletedDirectory(string path) {}     void up() {}     void down() {} }  notificationSession = manager.getDirectoryNotificationSessionForDirectory(myReceiver, "/Users/tom");</pre>
<b>Method</b>	<b>newDirectory</b>
<b>Description</b>	The callback for notifications of new directories.
<b>Parameters</b>	<b>string</b>   directoryPath        absolute pathname to the new directory
<b>Compliance</b>	<b>mandatory</b>   This method must be implemented.
<b>Method</b>	<b>changedDirectory</b>
<b>Description</b>	The callback for notification of updated directories.
<b>Parameters</b>	<b>string</b>   directoryPath        absolute pathname to the changed directory
<b>Compliance</b>	<b>mandatory</b>   This method must be implemented.
<b>Method</b>	<b>deletedDirectory</b>
<b>Description</b>	the callback for notification of deleted directories.
<b>Parameters</b>	<b>string</b>   directoryPath        absolute pathname to the deleted directory
<b>Compliance</b>	<b>mandatory</b>   This method must be implemented.

<b>Interface</b>	<b>osid.filing.DirectoryList</b>	
<b>Implements</b>	<a href="#">osid.OsidList</a>	
<b>Description</b>	<p>Like all OsidLists, DirectoryList provides a means for accessing Directory elements sequentially either one at a time or many at a time. Examples:</p> <pre>while (el.hasNext()) {     Entry entry = el.getNextEntry(); }</pre> <p>or</p> <pre>while (el.hasNext()) {     Entry[] entries = e</pre>	
<b>Method</b>	<b>getNextDirectory</b>	
<b>Description</b>	Gets the next Directory in this list.	
<b>Return</b>	<a href="#">osid.filing.Directory</a>	the next Directory in this list. The hasNext() method should be used to test that a next Dircectory is available before calling this method.
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a> <a href="#">OPERATION_FAILED</a>	no more elements available in this list unable to complete request
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.
<b>Method</b>	<b>getNextDirectories</b>	
<b>Description</b>	Gets the next set of Directory elements in this list which must be less than or equal to the number returned from available().	
<b>Parameters</b>	cardinal	n
		the number of Directory elements requested which should be less than or equal to available()
<b>Return</b>	<a href="#">osid.filing.Directory[]</a>	an array of Directory elements. The length of the array is less than or equal to the number specified.
<b>Errors</b>	<a href="#">ILLEGAL_STATE</a> <a href="#">OPERATION_FAILED</a>	no more elements available in this list unable to complete request
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.

<b>Interface</b>	<b>osid.filing.Allocation</b>	
<b>Implements</b>		
<b>Description</b>	Directory represents a directory in a file system that may contain other files and directories.	
<b>Method</b>	<b>getDirectoryPath</b>	
<b>Description</b>	Gets the absolute path to the directory responsible for these usage stats. The return may be a partition or volume containing the requested directory.	
<b>Return</b>	string	path name
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getDirectory</b>	
<b>Description</b>	Gets the directory responsible for these usage stats. The return may be a partition, quota controlled directory, or volume containing the requested directory.	
<b>Return</b>	osid.filing.Directory	directory
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getTotalSpace</b>	
<b>Description</b>	Gets the total space in bytes of this allocation.	
<b>Return</b>	cardinal	number of bytes in this allocation
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getUsedSpace</b>	
<b>Description</b>	Gets the used space in bytes of this allocation.	
<b>Return</b>	cardinal	number of used bytes in this allocation
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getAvailableSpace</b>	
<b>Description</b>	Gets the available space in bytes of this allocation.	
<b>Return</b>	cardinal	number of available bytes in this allocation
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getTotalFiles</b>	
<b>Description</b>	Gets the total number of files of this allocation.	
<b>Return</b>	cardinal	number of files in this allocation
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getUsedFiles</b>	
<b>Description</b>	Gets the used number of files of this allocation.	
<b>Return</b>	cardinal	number of used files in this allocation
<b>Compliance</b>	mandatory	This method must be implemented.
<b>Method</b>	<b>getAvailableFiles</b>	
<b>Description</b>	Gets the available number of files of this allocation.	
<b>Return</b>	cardinal	number of available files in this allocation
<b>Compliance</b>	mandatory	This method must be implemented.

<b>Interface</b>	<b>osid.filing.AllocationReceiver</b>				
<b>Implements</b>	<a href="#">osid.OsidReceiver</a>				
<b>Description</b>	The allocation receiver is the consumer supplied interface for receiving notifications pertaining to usage warnings and changed quota assignments.				
<b>Method</b>	<b>warning</b>				
<b>Description</b>	The callback for notifications of usage warnings.				
<b>Parameters</b>	<a href="#">string</a>	directoryPath	absolute pathname to the directory		
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.			
<b>Method</b>	<b>clearWarning</b>				
<b>Description</b>	The callback for notification of cleared usage warnings.				
<b>Parameters</b>	<a href="#">string</a>	directoryPath	absolute pathname to the directory		
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.			
<b>Method</b>	<b>changedQuota</b>				
<b>Description</b>	the callback for notification of a new, removed or changed quota assignment.				
<b>Parameters</b>	<a href="#">string</a>	directoryPath	absolute pathname to the directory		
<b>Compliance</b>	<a href="#">mandatory</a>	This method must be implemented.			