OSID V3 Specifications
transport package

Version Draft 3

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

Last Modified: 4 September 2008

prepared by:
Tom Coppeto
OnTapSolutions

	OSID License
	osid.OsidReceiver
Copyright	Copyright © 2002-2007 Massachusetts Institute of Technology. All Rights Reserved.
License	This Work is being provided by the copyright holder(s) subject to the following license. By obtaining, using and/or copying this Work, you agree that you have read, understand, and will comply with the following terms and conditions. This Work and the information contained herein is provided on an "AS IS" basis. The Massachusetts Institute of Technology, the Open Knowledge Initiative, and THE AUTHORS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE WORK OR THE USE OR OTHER DEALINGS IN THE WORK. Permission to use, copy and distribute unmodified versions of this Work, for any purpose, without fee or royalty is hereby granted, provided that you include the above copyright notice and the terms of this license on ALL copies of the Work or portions thereof. You may nodify or create Derivatives of this Work only for your internal purposes. You shall not distribute or transfer any such Derivative of this Work to any location or to any third party. For the purposes of this license, Derivative shall mean any derivative of the Work as defined in the United States Copyright Act of 1976, such as a translation or modification. The export of software employing encryption technology may require a specific license from the United States Government. It is the responsibility of any person or organization comtemplating export to obtain such a license before exporting this Work.

Package Description	osid.transport

Interfaces	osid.transport.TransportProfile osid.transport.TransportManager osid.transport.TransportSession osid.transport.TransportReceiver osid.transport.DataInputStream osid.transport.DataOutputStream

Package	**osid.transport**
Title	**Transport Open Service Interface Definitions**
Version	3.0.0

Description	The transport OSID provides a simple means of moving serialized data to or from the local endpoint. It defines to stream-like inerfaces, the DataInputStream and the DataOutputStream for reading and writing data.
	
	The transport OSID defines an inbound and an outbound service. An outbound transport service is used from a local endpoint to a single remote endpoint. The remote endpoint may represent one or more physical endpoints, such as a multicast network, but presents itself as a single enpoint to the consumer. The inbound service is used to receive and process data from multiple remote endpoints and is implemented using a callback mechanism when a new request for association arrives.
	An association or connection between two endpoints is represented by the TransportSession. Data may be written or read where each read or write is handled by a data stream. Ths transport OSID does not describe the format or sequence of the data which must be agreed upon at both ends. The OSID simply acts as a cover for specific transport or session related APIs, and provides a means for modular handling of service location for the purpose implemengting a redundancy or load balancing scheme. Structures and object-oriented data and presented through other OSIDs where applicable. The transport OSID may be used within another OSID to handle the transport of serialized data that may correspond to a higher level OSID object.
	
	The transport OSID can be used as the glue between an authentication OSID and the over-the-wire protocol that will be used to transport or receive a serialized authentication credential. A TransportSession may be instantiated by passing an authentication credential and conversely the authentication credential can be supplied to the transport receiver.
	The data streams defined in the transport OSID are also used in other OSIDs where a data stream is required such as the filing OSID or an Asset that contains arbitrary serialized data.
	Outbound example:
	`TransportSession session = manager.getOutboundSession();`
	`DataOutputStream out = session.sendData();`
	`String cmd = "<call><command>get_songs</command><params><param>" + album + "</param></call>";`
	`out.write(cmd.length(), cmd);`
	`out.close();`
	
	`while(session.hasDataAvailable()) {`
	`  DataInputStream in = session.receiveData();`
	`  add_song_to_play_queue(in);`
	`}`
	Inbound Example:

```
public void dispatch(TransportSession session, Authentication auth) {
    if (authzSession.isAuthorized(auth.getAgentId, fId, qId)) {
        String argv = parse_command_from_stream(session.receiverData());
        if (argv[0].equals("get_songs")) {
            Song songs = get_songs_from_album(argv[1]);
            for (song: songs) {
                DataOutputStream out = session.sendData();
                out.write(song.data(), song.size());
            }
        }
    }
}
```

Interface	**osid.transport.TransportProfile**	
Implements	osid.OsidProfile	
Description	The TransportProfile defines the interoperability of the transport OSID.	
Method	**supportsOutbound**	
Description	Tests if outbound transport is supported.	
Return	boolean	true if outbound transport is supported, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**supportsInbound**	
Description	Tests if inbound transport is supported.	
Return	boolean	true if incoming transport is supported, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**supportsAuthentication**	
Description	Tests if authentication is supported.	
Return	boolean	true if authentication is supported, false otherwise
Compliance	mandatory	This method must be implemented.

Interface	**osid.transport.TransportManager**		
Implements	osid.OsidManager		
	osid.transport.TransportProfile		
Description	This manager provides access to the sessions defined in this service. The outbound session is used to connect to a remote endpoint and the inbound callback is used to receive incoming connections.		
Method	**getOutboundSession**		
Description	Gets the OsidSession associated with the TransportOutbound service.		
Return	osid.transport.TransportSession		the new TransportOutboundSession
Errors	OPERATION_FAILED		unable to complete request
	UNIMPLEMENTED		supportsOutbound() is false
Compliance	optional		This method must be implemented if supportsOutbound() is true.
Method	**getOutboundSessionWithAuthentication**		
Description	Gets the OsidSession associated with the TransportOutbound service using the supplied Authentication.		
Parameters	osid.authentication.Authentication	authentication	proxy authentication
Return	osid.transport.TransportSession		the new TransportOutboundSession
Errors	NULL_ARGUMENT		authentication is null
	OPERATION_FAILED		unable to complete request
	PERMISSION_DENIED		the given authentication is invalid
	UNIMPLEMENTED		a transport outbound session is not available
	UNSUPPORTED		the authentication service is not supported
Compliance	optional		This method must be implemented if supportsOutbound() and supportsAuthentication() are true.
Method	**setInboundReceiver**		
Description	Sets the callback proc for incoming data.		
Parameters	osid.transport.TransportReceiver	receiver	the callback receiver
Errors	NULL_ARGUMENT		receiver is null
	OPERATION_FAILED		unable to complete request
	UNIMPLEMENTED		supportsInbound() is false
Compliance	optional		This method must be implemented if supportsInbound() is true.

Interface	**osid.transport.TransportSession**	
Implements	osid.OsidSession	
Description	The transport session is used to send and receive arbitrary data to and from a remote end point. The methods accept for return a data stream. Some protocols may send or receive all data within a single stream while others may use the streams as channels or frames of data. A stream may be available for reading before all the data as arrived and as such multiple streams may be processed simultaneously.	
Method	**sendData**	
Description	Sends data to the remote transport endpoint.	
Return	osid.transport.DataOutputStream	the output stream in which to send data
Errors	OPERATION_FAILED	unable to complete request
Compliance	mandatory	This method must be implemented.
Method	**hasDataAvailable**	
Description	Tests to see if another input stream is available for retrieval.	
Return	boolean	true if a stream is available for reading, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**receiveData**	
Description	Receives data from the remote transport endpoint.	
Return	osid.transport.DataInputStream	the input stream containing the received data
Errors	ILLEGAL_STATE	hasDataAvailable() is false
	OPERATION_FAILED	unable to complete request
Compliance	mandatory	This method must be implemented.

Interface	**osid.transport.TransportReceiver**		
Implements	osid.OsidReceiver		
Description	The TransportReceive is used to receive incoming connections. The receiver is provided to the service via the TransportManager and invoked by the transport provider when a new association is created. The transport session and authentication is porvided. The authentication object may contain information pertaining to the connection.		
Method	**dispatch**		
Description	Invoked by the transport provider when a new connection request or datagram is received.		
Parameters	osid.transport.TransportSession	session	the new transport session
	osid.authentication.Authentication	credential	the authentication credential retrieved from the transport or null if TransportManager.supportsPAuthenticationForProxy() is false
Compliance	mandatory		This method must be implemented.

Interface	**osid.transport.DataInputStream**		
Implements			
Description	The data input stream provides a means for reading data from a stream.		
Method	**atEndOfStream**		
Description	Tests if the end of this stream has been reached. This may not be a permanent condition as more data may be available at a later time as in the case of tailing a file.		
Return	boolean		true if the end of this stream has been reached, false otherwise
Errors	ILLEGAL_STATE		this stream has been closed
Compliance	mandatory		This method must be implemented.
Method	**available**		
Description	Gets the number of bytes available for retrieval. The number returned by this method may be less than or equal to the total number of bytes in this stream.		
Return	cardinal		the number of bytes available for retrieval
Errors	ILLEGAL_STATE		this stream has been closed
Compliance	mandatory		This method must be implemented.
Method	**skip**		
Description	Skips a specified number of bytes in the stream.		
Parameters	cardinal	n	the number of bytes to skip
Return	cardinal		the actual number of bytes skipped
Errors	ILLEGAL_STATE		this stream has been closed or atEndOfStream() is true
Compliance	mandatory		This method must be implemented.
Method	**read**		
Description	Reads a specified number of bytes from this stream.		
Parameters	byte[]	buf	the buffer in which the data is read
	cardinal	n	the number of bytes to read
Return	integer		the actual number of bytes read
Errors	ILLEGAL_STATE		this stream has been closed or atEndOfStream() is true
	INVALID_ARGUMENT		the size of buf is less than n
	NULL_ARGUMENT		buf is null
	OPERATION_FAILED		unable to complete request
Compliance	mandatory		This method must be implemented.
Method	**close**		
Description	Closes this stream and frees up any allocated resources. Methods in this object may not be invoked after this method is called.		
Errors	ILLEGAL_STATE		this stream has been closed
Compliance	mandatory		This method must be implemented.

Interface	**osid.transport.DataOutputStream**		
Implements			
Description	The data output stream provides a means in which data can be written to a stream.		
Method	**write**		
Description	Writes n bytes to this stream.		
Parameters	byte[]	buf	the buffer containing the data to write
	cardinal	n	the number of bytes to write
Errors	ILLEGAL_STATE		this stream has been closed
	INVALID_ARGUMENT		buf does not contain n bytes
	NULL_ARGUMENT		buf is null
	OPERATION_FAILED		unable to complete request
Compliance	mandatory		This method must be implemented.
Method	**close**		
Description	Flushes the output, closes this stream and frees up any allocated resources. Methods in this object may not be invoked after this method is called.		
Errors	ILLEGAL_STATE		this stream has been closed
Compliance	mandatory		This method must be implemented.