Tom Coppeto
Ingenescus

September 2013

Proxy of DisplayText Format Type

Status

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

Summary

The ability for an OSID Consumer to request a specific DisplayText format Type from OSID Providers through an OsidProxyManager requires an out of band agreement around a Proxy Record and ProxyConditionRecord Type. This is a request to proxy a format Type using core methods and have visibility into the format Type through OsidSessions.

Table of Contents
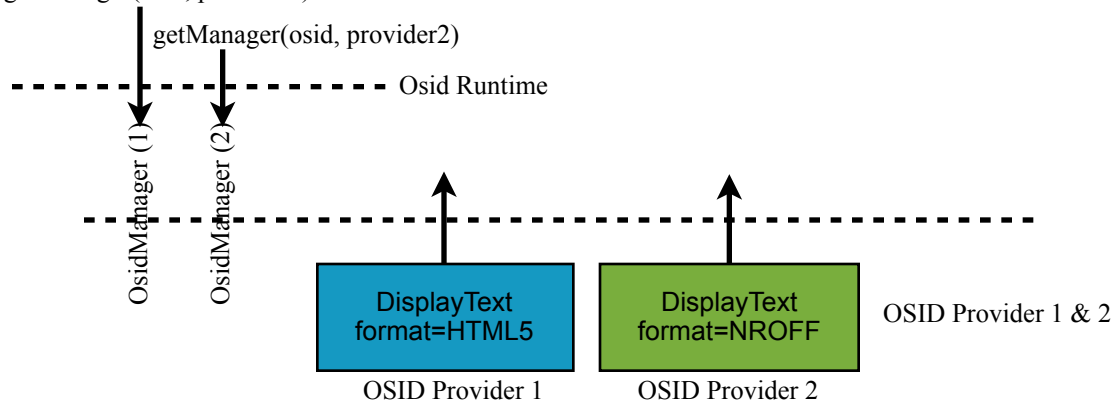
1. Current Specification

DisplayText is an OsidPrimitive to bundle a display string with a format Type. The format Type is an indicator to an OSID Consumer that is able to perform format conversions.
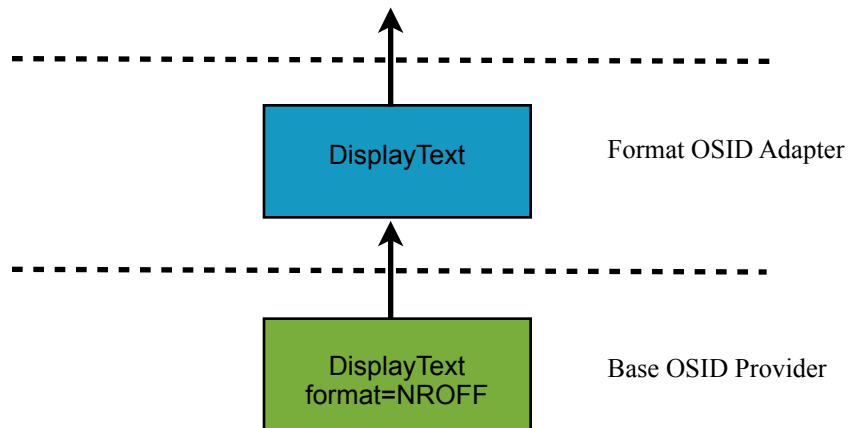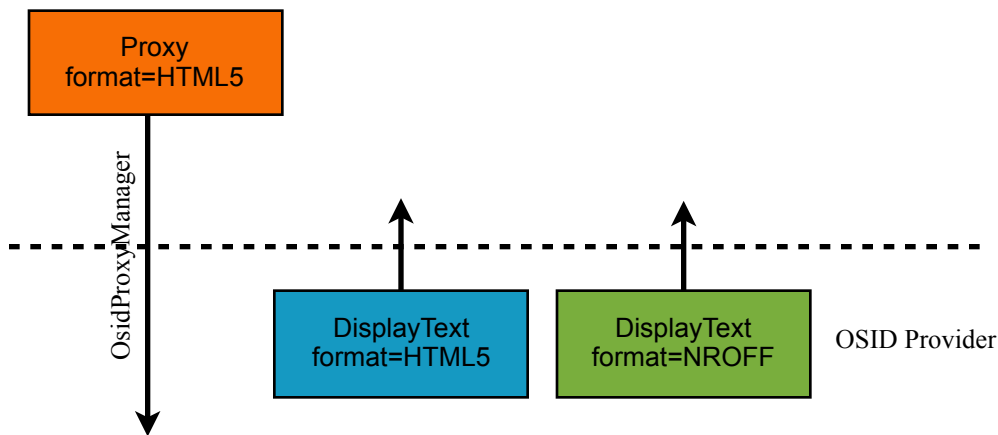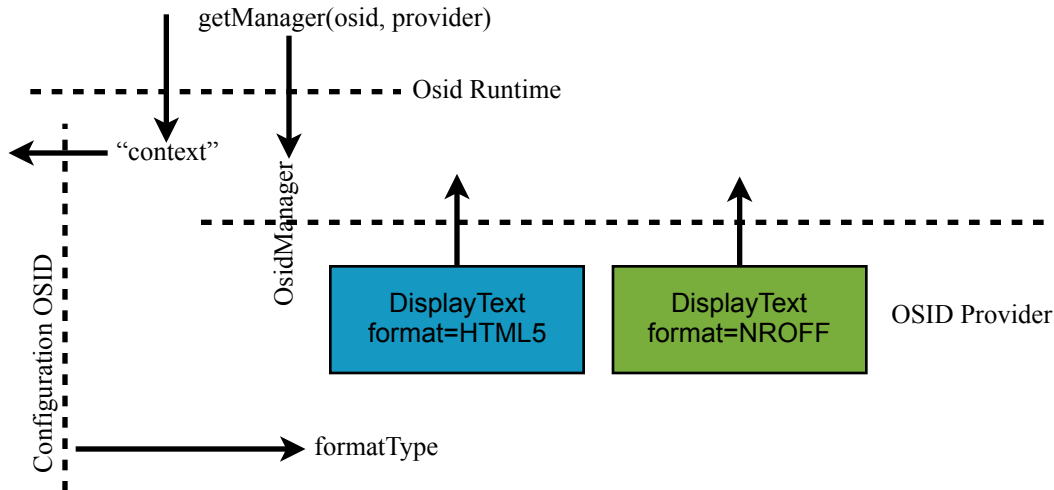

Alignment between of the DisplayText format between the application and the OSID Provider is an integration issue. For OSID Providers that may support more than one DisplayText format, the alignment may be performed by:

1. different service endpoints into the same OSID Provider implementation

2. configuration of the OSID Provider

getManager(osid, provider)

Osid Runtime

"context"

OsidManager

Configuration OSID

| DisplayText format=HTML5 | DisplayText format=NROFF | OSID Provider |

formatType

Proxy
format=HTML5

OsidProxyManager

| DisplayText format=HTML5 | DisplayText format=NROFF | OSID Provider |

3. in the case of a proxy scenario, a ProxyRecord defined to pass a format Type

2. Problem

1. While there is a core method to proxy locale and authentication information, there is no core method to proxy format to implement scenario 3 above. Currently, the format Type needs to be defined in a ProxyRecord for an OSID Provider that is able to support multiple formats. The definition of a format type in the OsidPrimitive DisplayText indicates that the issue of text formats is of *general concern*. Therefore, it is reasonable to include its explicit definition in the Proxy OSID where the Proxy is passed into OsidProxyManagers.

2. OsidSessions expose the Locale, effective date, and the effective agent. While this information may have been proxied by an OSID Consumer, the OSID Provider may or may not have honored it. The OsidSession provides the means to communicate what information actually in use by an OSID Provider. Similarly, the same should be true for the DisplayText format. Like the calendar and time Types, the OsidSession communicates the *mode* of the OSID Provider. It is conceivable that DisplayTexts of other format Types are seen as is the case with DateTime. The OSID specifications make no statement about such discrepancies.

3. Proposed Change

Add methods to osid.proxy.ProxyCondition and osid.proxy.Proxy to specify and access a format Type.

```
osid.proxy.ProxyCondition {
   ...

   /**
    *  Specifies the preferred DisplayText format Type.
    *
    *  @param formatType a DisplayText format Type
    *  @throws osid.NullArgumentException formatType is null
    */

   void setFormatType(osid.type.Type formatType);
}
```

```
osid.proxy.Proxy {
   ...

   /**
    *  Tests if a DisplayText format Type is available.
    *
    *  @return true if a format Type is available, false otherwise
    */

   boolean hasFormatType();

   /**
    *  Gets the preferred DisplayText format Type.
    *
    *  @return the preferred DisplayText format Type
    *  @throws osid.IllegalStateException hasFormatType() is false
    */

   osid.type.Type getFormatType();
}
```

Add a method to osid.OsidSession for a DisplayText format Type.

```
osid.OsidSession {
    ...

   /**
     *  Gets the DisplayText format Type preference in effect for
     *  this session.
     *
     *  @return the effective DisplayText format Type
     */

    osid.type.Type getFormatType();
}
```

4. Impacts

4.1. Specification

Changes are restricted to the addition of four new methods among osid.Proxy, osid.ProxyCondition, and osid.OsidSession.

4.2. OSID Consumers

No impacts on OSID Consumers since no interface methods are changed or removed. OSID Consumers may optionally make use of the format specifier in ProxyCondition or the format hint provided from OsidSessions..

4.3. OSID Providers

OSID Proxy Providers would be expected to minimally implement:

- ProxyCondition.setFormatType()
- Proxy.hasFormatType()
- OsidSession.getFormatType()

An OSID Provider supporting an OsidProxyManager would not required to use the DisplayText format Type in which case format level disagreements would be resolved at the provider level.

OsidSessions would be required to return a default format Type. Ideally, an OSID Provider should not flip back and forth among multiple DisplayText formats although there is nothing preventing it from doing so. It is also assumed that there is some default DisplayText format the OsidSession is delivering.

5. Interoperability Considerations

This change clarifies the specification for those OSID Providers supporting OsidProxyManagers how to switch between DisplayText formats if the preferred format

Type is present in the Proxy. The change eliminates the interoperability hurdle of a record Type agreement in the Proxy.

The presence or absence of the format Type in the Proxy is indicated by the `Proxy.hasFormatType()` method. This allows OSID Proxy Providers to easily support it or not support it.

The decision rests with an OSID Provider to honor the format Type, ignore the format Type, or supply a best effort match.

The only behavioral change requested is that OSID Providers seed OsidSessions with a default format Type.

6. Proposed Interfaces

6.1. osid.OsidSession

Interface	osid.OsidSession	
Implements		
Description	The OsidSession is the top level interface for all OSID sessions. An OsidSession is created through its corresponding OsidManager. A new OsidSession should be created for each user of a service and for each processing thread. A session maintains a single authenticated user and is not required to ensure thread-protection. A typical OSID session defines a set of service methods corresponding to some compliance level as defined by the service and is generally responsible for the management and retrieval of OsidObjects. OsidSession defines a set of common methods used throughout all OSID sessions. An OSID session may optionally support transactions through the transaction interface.	
Method	getLocale	
Description	Gets the locale indicating the localization preferences in effect for this session.	
Return	osid.locale.Locale	the locale
Compliance	mandatory	This method must be implemented.
Method	isAuthenticated	
Description	Tests if an agent is authenticated to this session.	
Return	boolean	true if valid authentication credentials exist, false otherwise
Compliance	mandatory	This method must be implemented.
Method	getAuthenticatedAgentId	
Description	Gets the Id of the agent authenticated to this session. This is the agent for which credentials are used either acquired natively or via an OsidProxyManager.	
Return	osid.id.Id	the authenticated agent Id
Errors	ILLEGAL_STATE	isAuthenticated() is false
Compliance	mandatory	This method must be implemented.
Method	getAuthenticatedAgent	
Description	Gets the agent authenticated to this session. This is the agent for which credentials are used either acquired natively or via an OsidProxyManager.	
Return	osid.authentication.Agent	the authenticated agent
Errors	ILLEGAL_STATE	isAuthenticated() is false
	OPERATION_FAILED	unable to complete request
Compliance	mandatory	This method must be implemented.

Method	getEffectiveAgentId	
Description	Gets the Id of the effective agent in use by this session. If isAuthenticated() is true, then the effective agent may be the same as the agent returned by getAuthenticatedAgent(). If isAuthenticated() is false, then the effective agent may be a default agent used for authorization by an unknwon or anonymous user.	
Return	osid.id.Id	the effective agent
Compliance	mandatory	This method must be implemented.

Method	getEffectiveAgent	
Description	Gets the effective agent in use by this session. If isAuthenticated() is true, then the effective agent may be the same as the agent returned by getAuthenticatedAgent(). If isAuthenticated() is false, then the effective agent may be a default agent used for authorization by an unknwon or anonymous user.	
Return	osid.authentication.Agent	the effective agent
Errors	OPERATION_FAILED	unable to complete request
Compliance	mandatory	This method must be implemented.

Method	getDate	
Description	Gets the service date which may be the current date or the effective date in which this session exists.	
Return	timestamp	the service date
Compliance	mandatory	This method must be implemented.

Method	getClockRate	
Description	Gets the rate of the service clock.	
Return	decimal	the clock rate
Compliance	mandatory	This method must be implemented.

Method	getFormatType	
Description	Gets the DisplayText format Type preference in effect for this session.	
Return	osid.type.Type	the effective DisplayText format Type
Compliance	mandatory	This method must be implemented.

Method	supportsTransactions	
Description	Tests for the availability of transactions.	
Return	boolean	true if transaction methods are available, false otherwise
Compliance	mandatory	This method must be implemented.

Method	startTransaction	
Description	Starts a new transaction for this sesson. Transactions are a means for an OSID to provide an all-or-nothing set of operations within a session and may be used to coordinate this service from an external transaction manager. A session supports one transaction at a time. Starting a second transaction before the previous has been committed or aborted results in an ILLEGAL_STATE error.	
Return	osid.transaction.Transaction	a new transaction
Errors	ILLEGAL_STATE	a transaction is already open
	OPERATION_FAILED	unable to complete request
	UNSUPPORTED	transactions not supported
Compliance	optional	This method must be implemented if supportsTransactions() is true.
Provider Notes	Ideally, a provider that supports transactions should guarantee atomicity, consistency, isolation and durability in a 2 phase commit process. This is not always possible in distributed systems and a transaction provider may simply allow for a means of processing bulk updates. To maximize interoperability, providers should honor the one-transaction-at-a-time rule.	

6.2.   osid.proxy.ProxyCondition

Interface	osid.proxy.ProxyCondition		
Implements	osid.OsidCondition		
Description	A ProxyCondition is used to transfer external information into a proxy.		
Method	setEffectiveAgentId		
Description	Sets the effective agent Id to indicate acting on behalf of.		
Parameters	osid.id.Id	agentId	an agent Id
Errors	NULL_ARGUMENT		agentId is null
Compliance	mandatory		This method must be implemented.
Method	setEffectiveDate		
Description	Sets the effective date.		
Parameters	timestamp	date	a date
	decimal	rate	the rate at which the clock should tick from the given effective date. 0 is a clock that is fixed
Errors	NULL_ARGUMENT		date is null
Compliance	mandatory		This method must be implemented.
Method	setLanguageType		
Description	Sets the language type.		
Parameters	osid.type.Type	languageType	the language type
Errors	NULL_ARGUMENT		languageType is null
Compliance	mandatory		This method must be implemented.
Method	setScriptType		
Description	Sets the script type.		
Parameters	osid.type.Type	scriptType	the script type
Errors	NULL_ARGUMENT		scriptType is null
Compliance	mandatory		This method must be implemented.
Method	setCalendarType		
Description	Sets the calendar type.		
Parameters	osid.type.Type	calendarType	the calendar type
Errors	NULL_ARGUMENT		calendarType is null
Compliance	mandatory		This method must be implemented.
Method	setTimeType		
Description	Sets the time type.		
Parameters	osid.type.Type	timeType	the time type
Errors	NULL_ARGUMENT		timeType is null
Compliance	mandatory		This method must be implemented.
Method	setCurrencyType		
Description	Sets the currency type.		
Parameters	osid.type.Type	currencyType	the currency type
Errors	NULL_ARGUMENT		currencyType is null
Compliance	mandatory		This method must be implemented.
Method	setUnitSystemType		
Description	Sets the unit system type.		
Parameters	osid.type.Type	unitSystemType	the unit system type
Errors	NULL_ARGUMENT		unitSystemType is null
Compliance	mandatory		This method must be implemented.
Method	setFormatType		
Description	Sets the DisplayText format type.		
Parameters	osid.type.Type	formatType	the format type
Errors	NULL_ARGUMENT		formatType is null

Compliance	mandatory	This method must be implemented.
Method	**getProxyConditionRecord**	
Description	Gets the proxy condition record corresponding to the given Proxy record Type. This method is used to retrieve an object implementing the requested record. The proxyRecordType may be the Type returned in getRecordTypes() or any of its parents in a Type hierarchy where hasRecordType(proxyRecordType) is true.	
Parameters	osid.type.Type	proxyConditionTyp	the type of proxy condition record to retrieve
Return	osid.proxy.records.ProxyConditionRecor	the proxy condition record
Errors	NULL_ARGUMENT	proxyConditionRecordType is null
	OPERATION_FAILED	unable to complete request
	UNSUPPORTED	hasRecordType(proxyConditionRecordType) is false
Compliance	mandatory	This method must be implemented.

6.3.   osid.proxy.Proxy

Interface	osid.proxy.Proxy	
Implements	osid.OsidResult	
Description	A Proxy is used to transfer external information from an application server into an OSID Provider.	
Method	**hasAuthentication**	
Description	Tests if an authentication is available.	
Return	boolean	true if an Authentication is available, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**getAuthentication**	
Description	Gets the Authentication for this proxy.	
Return	osid.authentication.process.Authenticati	the authentication
Errors	ILLEGAL_STATE	hasAuthentication() is false
Compliance	mandatory	This method must be implemented.
Method	**hasEffectiveAgent**	
Description	Tests if an effective agent is available.	
Return	boolean	true if an effective agent is available, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**getEffectiveAgentId**	
Description	Gets the effective Agent Id for this proxy.	
Return	osid.id.Id	the effective agent Id
Errors	ILLEGAL_STATE	hasEffectiveAgent() is false
Compliance	mandatory	This method must be implemented.
Method	**hasEffectiveDate**	
Description	Tests if an effective date is available.	
Return	boolean	true if an effective date is available, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**getEffectiveDate**	
Description	Gets the effective date.	
Return	timestamp	the effective date
Errors	ILLEGAL_STATE	hasEffectiveDate() is false

Compliance	mandatory	This method must be implemented.
Method	getEffectiveClockRate	
Description	Gets the rate of the clock.	
Return	decimal	the rate
Errors	ILLEGAL_STATE	hasEffectiveDate() is false
Compliance	mandatory	This method must be implemented.
Method	getLocale	
Description	Gets the locale.	
Return	osid.locale.Locale	a locale
Compliance	mandatory	This method must be implemented.
Method	hasFormatType	
Description	Tests if a DisplayText format Type is available.	
Return	boolean	true if a format type is available, false otherwise
Compliance	mandatory	This method must be implemented.
Method	getFormatType	
Description	Gets the DisplayText format Type.	
Return	osid.type.Type	the format Type
Errors	ILLEGAL_STATE	hasFormatType() is false
Compliance	mandatory	This method must be implemented.
Method	getProxyRecord	
Description	Gets the proxy record corresponding to the given Proxy record Type. This method is used to retrieve an object implementing the requested record. The proxyRecordType may be the Type returned in getRecordTypes() or any of its parents in a Type hierarchy where hasRecordType(proxyRecordType) is true.	
Parameters	osid.type.Type proxyRecordType	the type of proxy record to retrieve
Return	osid.proxy.records.ProxyRecord	the proxy record
Errors	NULL_ARGUMENT	proxyRecordType is null
	OPERATION_FAILED	unable to complete request
	UNSUPPORTED	hasRecordType(proxyRecordType) is false
Compliance	mandatory	This method must be implemented.

7. Copyright Statement