OSID Specification Request: 10
Track: Interface
Status: Proposed

Tom Coppeto
Ingenescus

July 2014

Managing DisplayTexts

Status

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

Summary

This is a specification change to use DisplayText in OsidForms where currently strings are specified. Using DisplayText allows for a clearer and explicit negotiation on an element by element basis as to the languages and formats permitted by an OSID Provider and supplied by an OSID Consumer.

Table of Contents

1. Current Specification

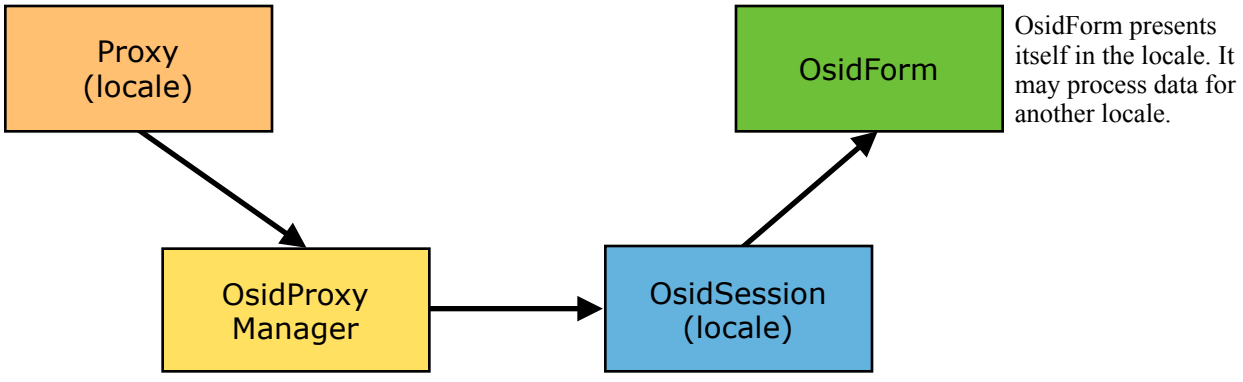
OsidForms that specify methods for DisplayText elements expect strings as parameters.

```
void setDisplayName(string displayName)
     throws InvalidAgument, NoAccess, NullArgument;
```

osid.locale.DisplayText is the following:

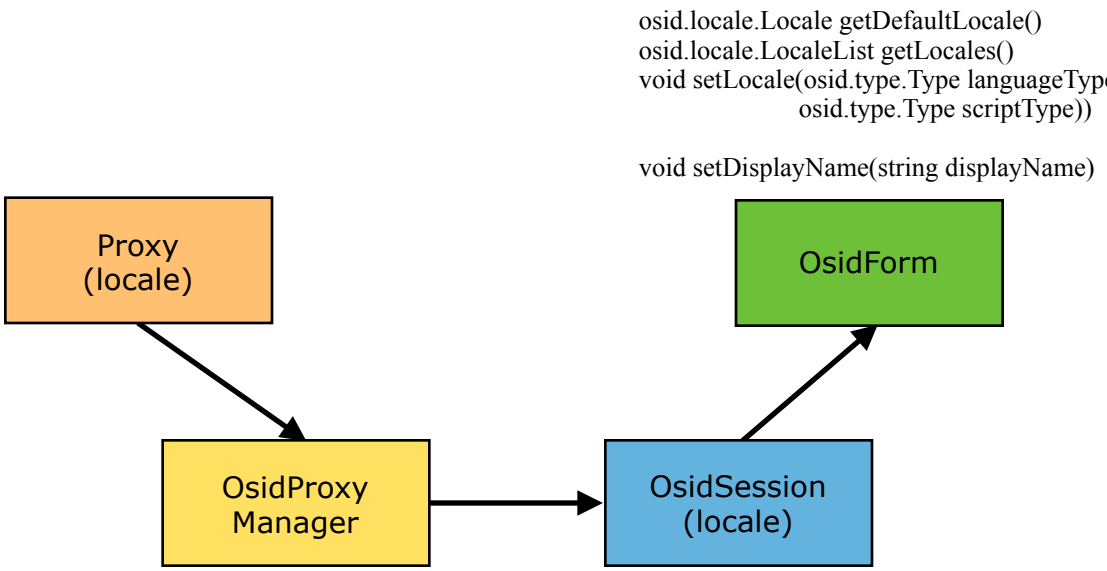
Interface	**osid.locale.DisplayText**	
Implements	osid.OsidPrimitive	
Description	Text to be displayed.	
Method	**getLanguageType**	
Description	Gets the language Type.	
Return	osid.type.Type	the language type
Compliance	mandatory	This method must be implemented.
Method	**getScriptType**	
Description	Gets the script Type.	
Return	osid.type.Type	the script type
Compliance	mandatory	This method must be implemented.
Method	**getFormatType**	
Description	Gets the format Type of the text string.	
Return	osid.type.Type	the format type
Compliance	mandatory	This method must be implemented.
Method	**getText**	
Description	Gets the text string.	
Return	string	the string
Compliance	mandatory	This method must be implemented.

The language and script Types are part of the localization mechanism. Locale is determined by an OSID Provider based on its configuration. An OSID Consumer may also supply hints through OsidProxyManagers.



This does not apply to the locale supplied into OsidForms. The administrative OsidSession may be in one locale while data of another locale is supplied into its OsidForms. This would be the case in a language translation scenario.

A language and script Type may currently be specified in an OsidForm presumably to inform the OSID Provider the intent of what language will be used. A single set of Types is used for all elements in the OsidForm.



Format Types are handled differently. The element's Metadata informs an OSID Consumer to the acceptable format Types. However, there is no means for an OSID Consumer to clarify which format is intended in the case multiple format Types are supported. OSID Providers are somehow expected to be able to discern among multiple format Types.

2. Problems

1. The constraints and specification of language, script, and format Types of DisplayText are inconsistent with the constraints and specification of other OsidPrimitives such as DateTime and Distance.

2. Setting language and script in an OsidForm allows for no variance on an element by element basis. Although this is not expected to be common, it may prove to be too restrictive in federated environments.

3. setLocale() in OsidForm is inconsistent with return interface from getLocales()

4. There is no means for specifying format Types for a multi-format OSID Provider. This is difficult if the formats are very similar.

5. Metadata does not use osid.Syntax.DISPLAYTEXT as DisplayTexts are conflated with strings.

3. Proposed Changes

3.1. Setting Strings in OsidForms

In every OsidForm method that accepts a string, change to accept a DisplayText.

3.2. osid.Metadata

Include the Type constraints and separate the metadata for osid.Syntax.DISPLAYTEXT and osid.Syntax.STRING.

Add:

- getDisplayTextLanguageTypes()
- supportsDisplayTextLanguageType()
- getDisplayTextScriptTypes()
- supportsDisplayTextScriptType()
- getDisplayTextFormatTypes()
- supportsDisplayTextFormatType()
- getDisplayTextSet()
- getDefaultDisplayTextValues()
- getExistingDisplayTextValues()

Remove:

- getStringFormatTypes()

The following methods apply to both DISPLAYTEXT and STRING:

- getMinimumStringLength()
- getMaximumStringLength()
- getStringMatchType()
- supportsStringMatchType()
- getStringExpression()

3.3. osid.OsidForm

Remove:

- getDefaultLocale()
- getLocales()
- setLocale()

4. Impacts

4.1. Specification

The changes outlined in 3.2 and 3.3 impact osid.Metadata and osid.OsidForm respectively. The changes outlined in 3.1 impact 53 methods in 30 OsidForms throughout the OSIDs.

4.2. OSID Consumers

This change impacts all OSID Consumers who manage strings which are visible through DisplayText.

4.3. OSID Providers

OSID Providers would be expected to support the new Metadata methods and distinguish between strings and DisplayTexts as needed.

5. Interoperability Considerations

The chief interoperability concern is the requirement that OSID Consumers supply a DisplayText instead of a string in OsidForms. Currently, an unspecified agreement exists between OSID Consumers and OSID Providers as to the locale and format of the string.

While the format is constrained to one of the formats in Metadata, OSID Providers can not definitively know which format was used. Also, the ability to specify locale in the OsidForm applies to all elements, but this assumes that all elements are localized in the same way.

This change treats DisplayText in a similar manner to other agreements such as DateTime, Version, and Distance where the constraints are explicitly conveyed through Metadata on a element by element basis, and the OSID Consumer is responsible for declaring which Types are in use by passing the complete OsidPrimitive.

On one hand, OSID Consumers will have to be actively part of the Type negotiation, but the agreements on locale and format are more explicit using Types.

The consistency achieved by this change extends to the semantics of an atomic OsidPrimitive that should not be specified in its component parts. It also improves the clarity in Metadata by clearly separating DisplayText from string.

6. Proposed Interfaces

6.1. osid.Metadata

Adds DisplayText methods.

Interface	**osid.Metadata**	
Implements		
Description	The Metadata interface defines a set of methods describing a the syntax and rules for creating and updating a data element inside an OsidForm. This interface provides a means to retrieve special restrictions placed upon data elements such as sizes and ranges that may vary from provider to provider or from object to object.	
Method	**getElementId**	
Description	Gets a unique Id for the data element.	
Return	osid.id.Id	an Id
Compliance	mandatory	This method must be implemented.
Method	**getElementLabel**	
Description	Gets a display label for the data element.	
Return	osid.locale.DisplayText	a display label
Compliance	mandatory	This method must be implemented.
Method	**getInstructions**	
Description	Gets instructions for updating this element value. This is a human readable description of the data element or property that may include special instructions or caveats to the end-user above and beyond what this interface provides.	
Return	osid.locale.DisplayText	instructions
Compliance	mandatory	This method must be implemented.
Method	**getSyntax**	
Description	Gets the syntax of this data.	
Return	osid.Syntax	an enumeration indicating the type of value
Compliance	mandatory	This method must be implemented.
Method	**isArray**	
Description	Tests if this data element is an array.	
Return	boolean	true if this data is an array, false if a single element
Compliance	mandatory	This method must be implemented.
Method	**isRequired**	
Description	Tests if this data element is required for creating new objects.	
Return	boolean	true if this element value is required, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**isReadOnly**	
Description	Tests if this data can be updated. This may indicate the result of a pre-authorization but is not a guarantee that an authorization failure will not occur when the create or update transaction is issued.	
Return	boolean	true if this data is not updatable, false otherwise
Compliance	mandatory	This method must be implemented.
Method	**isLinked**	
Description	Tests if this data element is linked to other data in the object. Updating linked data elements should refresh all metadata and revalidate object elements.	

Return	boolean	true if this element is linked, false if updates have no side effect
Compliance	mandatory	This method must be implemented.
Method	**isValueKnown**	
Description	Tests if an existing value is known for this data element. If it is known that a value does not exist, then this method returns true.	
Return	boolean	true if the element value is known, false if the element value is not known
Compliance	mandatory	This method must be implemented.
Method	**hasValue**	
Description	Tests if this data element has a set non-default value.	
Return	boolean	true if this element value has been set, false otherwise
Errors	ILLEGAL_STATE	isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getUnits**	
Description	Gets the units of this data for display purposes ('lbs', 'gills', 'furlongs').	
Return	osid.locale.DisplayText	the display units of this data or an empty string if not applicable
Compliance	mandatory	This method must be implemented.
Method	**getMinimumElements**	
Description	In the case where an array or list of elements is specified in an OsidForm, this specifies the minimum number of elements that must be included.	
Return	cardinal	the minimum elements or 1 if isArray() is false
Compliance	mandatory	This method must be implemented.
Method	**getMaximumElements**	
Description	In the case where an array or list of elements is specified in an OsidForm, this specifies the maximum number of elements that can be specified.	
Return	cardinal	the maximum elements or 1 if isArray() is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumCardinal**	
Description	Gets the minimum cardinal value.	
Return	cardinal	the minimum cardinal
Errors	ILLEGAL_STATE	syntax is not a CARDINAL
Compliance	mandatory	This method must be implemented.
Method	**getMaximumCardinal**	
Description	Gets the maximum cardinal value.	
Return	cardinal	the maximum cardinal
Errors	ILLEGAL_STATE	syntax is not a CARDINAL
Compliance	mandatory	This method must be implemented.
Method	**getCardinalSet**	
Description	Gets the set of acceptable cardinal values.	
Return	cardinal[]	a set of cardinals or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a CARDINAL
Compliance	mandatory	This method must be implemented.
Method	**getDefaultCardinalValues**	
Description	Gets the default cardinal values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	cardinal[]	the default cardinal values

Errors	ILLEGAL_STATE	syntax is not a CARDINAL or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingCardinalValues**	
Description	Gets the existing cardinal values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	cardinal[]	the existing cardinal values
Errors	ILLEGAL_STATE	syntax is not a CARDINAL or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getCoordinateTypes**	
Description	Gets the set of acceptable coordinate types.	
Return	osid.type.Type[]	the set of coordinate types
Errors	ILLEGAL_STATE	syntax is not a COORDINATE or SPATIALUNIT
Compliance	mandatory	This method must be implemented.
Method	**supportsCoordinateType**	
Description	Tests if the given coordinate type is supported.	
Parameters	osid.type.Type coordinateType	a coordinate Type
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not a COORDINATE
	NULL_ARGUMENT	coordinateType is null
Compliance	mandatory	This method must be implemented.
Method	**getAxesForCoordinateType**	
Description	Gets the number of axes for a given supported coordinate type.	
Parameters	osid.type.Type coordinateType	a coordinate Type
Return	cardinal	the number of axes
Errors	ILLEGAL_STATE	syntax is not a COORDINATE
	NULL_ARGUMENT	coordinateType is null
	UNSUPPORTED	supportsCoordinateType(coordinateType) is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumCoordinateValues**	
Description	Gets the minimum coordinate values given supported coordinate type.	
Parameters	osid.type.Type coordinateType	a coordinate Type
Return	decimal[]	the minimum coordinate values
Errors	ILLEGAL_STATE	syntax is not a COORDINATE
	NULL_ARGUMENT	coordinateType is null
	UNSUPPORTED	supportsCoordinateType(coordinateType) is false
Compliance	mandatory	This method must be implemented.
Method	**getMaximumCoordinateValues**	
Description	Gets the maximum coordinate values given supported coordinate type.	
Parameters	osid.type.Type coordinateType	a coordinate Type
Return	decimal[]	the maximum coordinate values
Errors	ILLEGAL_STATE	syntax is not a COORDINATE
	NULL_ARGUMENT	coordinateType is null
	UNSUPPORTED	supportsCoordinateType(coordinateType) is false
Compliance	mandatory	This method must be implemented.
Method	**getCoordinateSet**	

Description	Gets the set of acceptable coordinate values.	
Return	osid.mapping.Coordinate[]	a set of coordinates or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a COORDINATE
Compliance	mandatory	This method must be implemented.
Method	**getDefaultCoordinateValues**	
Description	Gets the default coordinate values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.Coordinate[]	the default coordinate values
Errors	ILLEGAL_STATE	syntax is not a COORDINATE or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingCoordinateValues**	
Description	Gets the existing coordinate values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.Coordinate[]	the existing coordinate values
Errors	ILLEGAL_STATE	syntax is not a COORDINATE or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getCurrencyTypes**	
Description	Gets the set of acceptable currency types.	
Return	osid.type.Type[]	the set of currency types
Errors	ILLEGAL_STATE	syntax is not a CURRENCY
Compliance	mandatory	This method must be implemented.
Method	**supportsCurrencyType**	
Description	Tests if the given currency type is supported.	
Parameters	osid.type.Type currencyType	a currency Type
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not a CURRENCY
	NULL_ARGUMENT	currencyType is null
Compliance	mandatory	This method must be implemented.
Method	**getMinimumCurrency**	
Description	Gets the minimum currency value.	
Return	osid.financials.Currency	the minimum currency
Errors	ILLEGAL_STATE	syntax is not a CURRENCY
Compliance	mandatory	This method must be implemented.
Method	**getMaximumCurrency**	
Description	Gets the maximum currency value.	
Return	osid.financials.Currency	the maximum currency
Errors	ILLEGAL_STATE	syntax is not a CURRENCY
Compliance	mandatory	This method must be implemented.
Method	**getCurrencySet**	
Description	Gets the set of acceptable currency values.	
Return	osid.financials.Currency[]	a set of currencies or an empty array if not
Errors	ILLEGAL_STATE	syntax is not a CURRENCY
Compliance	mandatory	This method must be implemented.
Method	**getDefaultCurrencyValues**	
Description	Gets the default currency values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.financials.Currency[]	the default currency values

Errors	ILLEGAL_STATE		syntax is not a CURRENCY or isRequired() is true
Compliance	mandatory		This method must be implemented.
Method	**getExistingCurrencyValues**		
Description	Gets the existing currency values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.		
Return	osid.financials.Currency[]		the existing currency values
Errors	ILLEGAL_STATE		syntax is not a CURRENCY or isValueKnown() is false
Compliance	mandatory		This method must be implemented.
Method	**getDateTimeResolution**		
Description	Gets the smallest resolution of the date time value.		
Return	osid.calendaring.DateTimeResolution		the resolution
Errors	ILLEGAL_STATE		syntax is not a DATETIME, DURATION, or TIME
Compliance	mandatory		This method must be implemented.
Method	**getCalendarTypes**		
Description	Gets the set of acceptable calendar types.		
Return	osid.type.Type[]		the set of calendar types
Errors	ILLEGAL_STATE		syntax is not a DATETIME or DURATION
Compliance	mandatory		This method must be implemented.
Method	**supportsCalendarType**		
Description	Tests if the given calendar type is supported.		
Parameters	osid.type.Type	calendarType	a calendar Type
Return	boolean		true if the type is supported, false otherwise
Errors	ILLEGAL_STATE		syntax is not a DATETIME or DURATION
	NULL_ARGUMENT		calendarType is null
Compliance	mandatory		This method must be implemented.
Method	**getTimeTypes**		
Description	Gets the set of acceptable time types.		
Return	osid.type.Type[]		a set of time types or an empty array if not restricted
Errors	ILLEGAL_STATE		syntax is not a DATETIME, DURATION, or TIME
Compliance	mandatory		This method must be implemented.
Method	**supportsTimeType**		
Description	Tests if the given time type is supported.		
Parameters	osid.type.Type	timeType	a time Type
Return	boolean		true if the type is supported, false otherwise
Errors	ILLEGAL_STATE		syntax is not a DATETIME, DURATION, or TIME
	NULL_ARGUMENT		timeType is null
Compliance	mandatory		This method must be implemented.
Method	**getMinimumDateTime**		
Description	Gets the minimum date time value.		
Return	osid.calendaring.DateTime		the minimum value
Errors	ILLEGAL_STATE		syntax is not a DATETIME
Compliance	mandatory		This method must be implemented.
Method	**getMaximumDateTime**		
Description	Gets the maximum date time value.		
Return	osid.calendaring.DateTime		the maximum value

Errors	ILLEGAL_STATE	syntax is not a DATETIME
Compliance	mandatory	This method must be implemented.
Method	**getDateTimeSet**	
Description	Gets the set of acceptable date time values.	
Return	osid.calendaring.DateTime[]	a set of values or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a DATETIME
Compliance	mandatory	This method must be implemented.
Method	**getDefaultDateTimeValues**	
Description	Gets the default date time values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.calendaring.DateTime[]	the default date time values
Errors	ILLEGAL_STATE	syntax is not a DATETIME or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingDateTimeValues**	
Description	Gets the existing date time values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.calendaring.DateTime[]	the existing date time values
Errors	ILLEGAL_STATE	syntax is not a DATETIME or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getDecimalScale**	
Description	Gets the number of digits to the right of the decimal point.	
Return	cardinal	the scale
Errors	ILLEGAL_STATE	syntax is not a DECIMAL
Compliance	mandatory	This method must be implemented.
Method	**getMinimumDecimal**	
Description	Gets the minimum decimal value.	
Return	decimal	the minimum decimal
Errors	ILLEGAL_STATE	syntax is not a DECIMAL
Compliance	mandatory	This method must be implemented.
Method	**getMaximumDecimal**	
Description	Gets the maximum decimal value.	
Return	decimal	the maximum decimal
Errors	ILLEGAL_STATE	syntax is not a DECIMAL
Compliance	mandatory	This method must be implemented.
Method	**getDecimalSet**	
Description	Gets the set of acceptable decimal values.	
Return	decimal[]	a set of decimals or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a DECIMAL
Compliance	mandatory	This method must be implemented.
Method	**getDefaultDecimalValues**	
Description	Gets the default decimal values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	decimal[]	the default decimal values
Errors	ILLEGAL_STATE	syntax is not a DECIMAL or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingDecimalValues**	

Description	Gets the existing decimal values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.		
Return	decimal[]		the existing decimal values
Errors	ILLEGAL_STATE		syntax is not a DECIMAL or isValueKnown() is false
Compliance	mandatory		This method must be implemented.
Method	**getDisplayTextLanguageTypes**		
Description	Gets the set of acceptable language types.		
Return	osid.type.Type[]		the set of language types
Errors	ILLEGAL_STATE		syntax is not a DISPLAYTEXT
Compliance	mandatory		This method must be implemented.
Method	**supportsDisplayTextLanguageType**		
Description	Tests if the given language type is supported.		
Parameters	osid.type.Type	languageType	a language Type
Return	boolean		true if the type is supported, false otherwise
Errors	ILLEGAL_STATE		syntax is not a DISPLAYTEXT
	NULL_ARGUMENT		languageType is null
Compliance	mandatory		This method must be implemented.
Method	**getDisplayTexttScriptTypes**		
Description	Gets the set of acceptable script types for a language type.		
Parameters	osid.type.Type	languageType	a language Type
Return	osid.type.Type[]		the set of script types
Errors	ILLEGAL_STATE		syntax is not a DISPLAYTEXT
	NULL_ARGUMENT		languageType is null
	UNSUPPORTED		supportsLanguageType(languageType) is false
Compliance	mandatory		This method must be implemented.
Method	**supportsDisplayTextScriptType**		
Description	Tests if the given script type is supported.		
Parameters	osid.type.Type	languageType	a language Type
	osid.type.Type	scriptType	a script Type
Return	boolean		true if the type is supported, false otherwise
Errors	ILLEGAL_STATE		syntax is not a DISPLAYTEXT
	NULL_ARGUMENT		languageType or scriptType is null
Compliance	mandatory		This method must be implemented.
Method	**getDisplayTexttFormatTypes**		
Description	Gets the set of acceptable format types.		
Return	osid.type.Type[]		the set of format types
Errors	ILLEGAL_STATE		syntax is not a DISPLAYTEXT
Compliance	mandatory		This method must be implemented.
Method	**supportsDisplayTextFormatType**		
Description	Tests if the given format type is supported.		
Parameters	osid.type.Type	formatType	a format Type
Return	boolean		true if the type is supported, false otherwise
Errors	ILLEGAL_STATE		syntax is not a DISPLAYTEXT
	NULL_ARGUMENT		formatType is null
Compliance	mandatory		This method must be implemented.
Method	**getDisplayTextSet**		
Description	Gets the set of acceptable distance values.		
Return	osid.locale.DisplayText[]		a set of values or an empty array if not restricted
Errors	ILLEGAL_STATE		syntax is not a DISTANCE

Compliance	mandatory	This method must be implemented.
Method	**getDefaultDisplayTextValues**	
Description	Gets the default text values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.locale.DisplayText[]	the default text values
Errors	ILLEGAL_STATE	syntax is not a DISPLAYTEXT or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingDisplayTextValues**	
Description	Gets the existing text values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.locale.DisplayText[]	the existing distance values
Errors	ILLEGAL_STATE	syntax is not a DISPLAYTEXT or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getDistanceResolution**	
Description	Gets the smallest resolution of the distance value.	
Return	osid.mapping.DistanceResolution	the resolution
Errors	ILLEGAL_STATE	syntax is not a DISTANCE
Compliance	mandatory	This method must be implemented.
Method	**getMinimumDistance**	
Description	Gets the minimum distance value.	
Return	osid.mapping.Distance	the minimum value
Errors	ILLEGAL_STATE	syntax is not a DISTANCE
Compliance	mandatory	This method must be implemented.
Method	**getMaximumDistance**	
Description	Gets the maximum distance value.	
Return	osid.mapping.Distance	the maximum value
Errors	ILLEGAL_STATE	syntax is not a DISTANCE
Compliance	mandatory	This method must be implemented.
Method	**getDistanceSet**	
Description	Gets the set of acceptable distance values.	
Return	osid.mapping.Distance[]	a set of values or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a DISTANCE
Compliance	mandatory	This method must be implemented.
Method	**getDefaultDistanceValues**	
Description	Gets the default distance values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.Distance[]	the default distance values
Errors	ILLEGAL_STATE	syntax is not a DISTANCE or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingDistanceValues**	
Description	Gets the existing distance values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.Distance[]	the existing distance values
Errors	ILLEGAL_STATE	syntax is not a DISTANCE or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumDuration**	
Description	Gets the minimum duration.	

Return	osid.calendaring.Duration	the minimum duration
Errors	ILLEGAL_STATE	syntax is not a DURATION
Compliance	mandatory	This method must be implemented.
Method	**getMaximumDuration**	
Description	Gets the maximum duration.	
Return	osid.calendaring.Duration	the maximum duration
Errors	ILLEGAL_STATE	syntax is not a DURATION
Compliance	mandatory	This method must be implemented.
Method	**getDurationSet**	
Description	Gets the set of acceptable duration values.	
Return	osid.calendaring.Duration[]	a set of durations or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a DURATION
Compliance	mandatory	This method must be implemented.
Method	**getDefaultDurationValues**	
Description	Gets the default duration values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most at most a single value.	
Return	osid.calendaring.Duration[]	the default duration values
Errors	ILLEGAL_STATE	syntax is not a DURATION or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingDurationValues**	
Description	Gets the existing duration values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.calendaring.Duration[]	the existing duration values
Errors	ILLEGAL_STATE	syntax is not a DURATION or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getHeadingTypes**	
Description	Gets the set of acceptable heading types.	
Return	osid.type.Type[]	a set of heading types or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a HEADING
Compliance	mandatory	This method must be implemented.
Method	**supportsHeadingType**	
Description	Tests if the given heading type is supported.	
Parameters	osid.type.Type | headingType	a heading Type
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not a HEADING
	NULL_ARGUMENT	headingType is null
Compliance	mandatory	This method must be implemented.
Method	**getAxesForHeadingType**	
Description	Gets the number of axes for a given supported heading type.	
Parameters	osid.type.Type | headingType	a heading Type
Return	cardinal	the number of axes
Errors	ILLEGAL_STATE	syntax is not a HEADING
	NULL_ARGUMENT	headingType is null
	UNSUPPORTED	supportsHeadingType(headingType) is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumHeadingValues**	
Description	Gets the minimum heading values given supported heading type.	

Parameters	osid.type.Type	headingType	a heading Type
Return	decimal[]		the minimum heading values
Errors	ILLEGAL_STATE		syntax is not a HEADING
	NULL_ARGUMENT		headingType is null
	UNSUPPORTED		supportsHeadingType(headingType) is false
Compliance	mandatory		This method must be implemented.
Method	**getMaximumHeadingValues**		
Description	Gets the maximum heading values given supported heading type.		
Parameters	osid.type.Type	headingType	a heading Type
Return	decimal[]		the maximum heading values
Errors	ILLEGAL_STATE		syntax is not a HEADING
	NULL_ARGUMENT		headingType is null
	UNSUPPORTED		supportsHeadingType(headingType) is false
Compliance	mandatory		This method must be implemented.
Method	**getHeadingSet**		
Description	Gets the set of acceptable heading values.		
Return	osid.mapping.Heading[]		the set of heading
Errors	ILLEGAL_STATE		syntax is not a HEADING
Compliance	mandatory		This method must be implemented.
Method	**getDefaultHeadingValues**		
Description	Gets the default heading values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.		
Return	osid.mapping.Heading[]		the default heading values
Errors	ILLEGAL_STATE		syntax is not a HEADING or isRequired() is true
Compliance	mandatory		This method must be implemented.
Method	**getExistingHeadingValues**		
Description	Gets the existing heading values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.		
Return	osid.mapping.Heading[]		the existing heading values
Errors	ILLEGAL_STATE		syntax is not a HEADING or isValueKnown() is false
Compliance	mandatory		This method must be implemented.
Method	**getIdSet**		
Description	Gets the set of acceptable Ids.		
Return	osid.id.Id[]		a set of Ids or an empty array if not restricted
Errors	ILLEGAL_STATE		syntax is not an ID
Compliance	mandatory		This method must be implemented.
Method	**getDefaultIdValues**		
Description	Gets the default Id values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.		
Return	osid.id.Id[]		the default Id values
Errors	ILLEGAL_STATE		syntax is not an ID or isRequired() is true
Compliance	mandatory		This method must be implemented.
Method	**getExistingIdValues**		
Description	Gets the existing Id values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.		
Return	osid.id.Id[]		the existing Id values
Errors	ILLEGAL_STATE		syntax is not an ID
Compliance	mandatory		This method must be implemented.
Method	**getMinimumInteger**		

Description	Gets the minimum integer value.	
Return	integer	the minimum value
Errors	ILLEGAL_STATE	syntax is not an INTEGER
Compliance	mandatory	This method must be implemented.
Method	**getMaximumInteger**	
Description	Gets the maximum integer value.	
Return	integer	the maximum value
Errors	ILLEGAL_STATE	syntax is not an INTEGER
Compliance	mandatory	This method must be implemented.
Method	**getIntegerSet**	
Description	Gets the set of acceptable integer values.	
Return	integer[]	a set of values or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not an INTEGER
Compliance	mandatory	This method must be implemented.
Method	**getDefaultIntegerValues**	
Description	Gets the default integer values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	integer[]	the default integer values
Errors	ILLEGAL_STATE	syntax is not an INTEGER or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingIntegerValues**	
Description	Gets the existing integer values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	integer[]	the existing integer values
Errors	ILLEGAL_STATE	syntax is not a INTEGER or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getObjectTypes**	
Description	Gets the set of acceptable Types for an arbitrary object.	
Return	osid.type.Type[]	a set of Types or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not an OBJECT
Compliance	mandatory	This method must be implemented.
Method	**supportsObjectType**	
Description	Tests if the given object type is supported.	
Parameters	osid.type.Type objectType	an object Type
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not an OBJECT
	NULL_ARGUMENT	objectType is null
Compliance	mandatory	This method must be implemented.
Method	**getObjectSet**	
Description	Gets the set of acceptable object values.	
Return	object[]	a set of values or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not an OBJECT
Compliance	mandatory	This method must be implemented.
Method	**getDefaultObjectValues**	
Description	Gets the default object values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	object[]	the default object values
Errors	ILLEGAL_STATE	syntax is not an OBJECT or isRequired() is true

Compliance	mandatory	This method must be implemented.
Method	**getExistingObjectValues**	
Description	Gets the existing object values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	object[]	the existing object values
Errors	ILLEGAL_STATE	syntax is not an OBJECT or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getSpatialUnitRecordTypes**	
Description	Gets the set of acceptable spatial unit record types.	
Return	osid.type.Type[]	the set of spatial unit types
Errors	ILLEGAL_STATE	syntax is not SPATIALUNIT
Compliance	mandatory	This method must be implemented.
Method	**supportsSpatialUnitRecordType**	
Description	Tests if the given spatial unit record type is supported.	
Parameters	osid.type.Type spatialUnitRecordType	a spatial unit record Type
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not an SPATIALUNIT
	NULL_ARGUMENT	spatialUnitRecordType is null
Compliance	mandatory	This method must be implemented.
Method	**getSpatialUnitSet**	
Description	Gets the set of acceptable spatial unit values.	
Return	osid.mapping.SpatialUnit[]	a set of spatial units or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a SPATIALUNIT
Compliance	mandatory	This method must be implemented.
Method	**getDefaultSpatialUnitValues**	
Description	Gets the default spatial unit values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.SpatialUnit[]	the default spatial unit values
Errors	ILLEGAL_STATE	syntax is not a SPATIALUNIT or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingSpatialUnitValues**	
Description	Gets the existing spatial unit values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single	
Return	osid.mapping.SpatialUnit[]	the existing spatial unit values
Errors	ILLEGAL_STATE	syntax is not a SPATIALUNIT or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumSpeed**	
Description	Gets the minimum speed value.	
Return	osid.mapping.Speed	the minimum speed
Errors	ILLEGAL_STATE	syntax is not a SPEED
Compliance	mandatory	This method must be implemented.
Method	**getMaximumSpeed**	
Description	Gets the maximum speed value.	
Return	osid.mapping.Speed	the maximum speed
Errors	ILLEGAL_STATE	syntax is not a SPEED
Compliance	mandatory	This method must be implemented.
Method	**getSpeedSet**	

Description	Gets the set of acceptable speed values.	
Return	osid.mapping.Speed[]	a set of speeds or an empty array if not
Errors	ILLEGAL_STATE	syntax is not a SPEED
Compliance	mandatory	This method must be implemented.
Method	**getDefaultSpeedValues**	
Description	Gets the default speed values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.Speed[]	the default speed values
Errors	ILLEGAL_STATE	syntax is not a SPEED or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingSpeedValues**	
Description	Gets the existing speed values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.mapping.Speed[]	the existing speed values
Errors	ILLEGAL_STATE	syntax is not a SPEED or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumStringLength**	
Description	Gets the minimum string length.	
Return	cardinal	the minimum string length
Errors	ILLEGAL_STATE	syntax is not a STRING or DISPLAYTEXT
Compliance	mandatory	This method must be implemented.
Method	**getMaximumStringLength**	
Description	Gets the maximum string length.	
Return	cardinal	the maximum string length
Errors	ILLEGAL_STATE	syntax is not a STRING or DISPLAYTEXT
Compliance	mandatory	This method must be implemented.
Method	**getStringMatchTypes**	
Description	Gets the set of valid string match types for use in validating a string. If the string match type indicates a regular expression then getStringExpression() returns a regular expression.	
Return	osid.type.Type[]	the set of string match types
Errors	ILLEGAL_STATE	syntax is not a STRING or DISPLAYTEXT
Compliance	mandatory	This method must be implemented.
Method	**supportsStringMatchType**	
Description	Tests if the given string match type is supported.	
Parameters	osid.type.Type stringMatchType	a string match type
Return	boolean	true if the given string match type Is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not a STRING or DISPLAYTEXT
	NULL_ARGUMENT	stringMatchType is null
Compliance	mandatory	This method must be implemented.
Method	**getStringExpression**	
Description	Gets the regular expression of an acceptable string for the given string match type.	
Parameters	osid.type.Type stringMatchType	a string match type
Return	string	the regular expression
Errors	NULL_ARGUMENT	stringMatchType is null
	ILLEGAL_STATE	syntax is not a STRING or DISPLAYTEXT
	UNSUPPORTED	supportsStringMatchType(stringMatchType) is false
Compliance	mandatory	This method must be implemented.
Method	**getStringSet**	

Description	Gets the set of acceptable string values.	
Return	string[]	a set of strings or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a STRING
Compliance	mandatory	This method must be implemented.
Method	**getDefaultStringValues**	
Description	Gets the default string values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	string[]	the default string values
Errors	ILLEGAL_STATE	syntax is not a STRING or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingStringValues**	
Description	Gets the existing string values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	string[]	the existing string values
Errors	ILLEGAL_STATE	syntax is not a STRING or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getMinimumTime**	
Description	Gets the minimum time value.	
Return	osid.calendaring.Time	the minimum time
Errors	ILLEGAL_STATE	syntax is not a TIME
Compliance	mandatory	This method must be implemented.
Method	**getMaximumTime**	
Description	Gets the maximum time value.	
Return	osid.calendaring.Time	the maximum time
Errors	ILLEGAL_STATE	syntax is not a TIME
Compliance	mandatory	This method must be implemented.
Method	**getTimeSet**	
Description	Gets the set of acceptable time values.	
Return	osid.calendaring.Time[]	a set of times or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a TIME
Compliance	mandatory	This method must be implemented.
Method	**getDefaultTimeValues**	
Description	Gets the default time values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.calendaring.Time[]	the default time values
Errors	ILLEGAL_STATE	syntax is not a TIME or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingTimeValues**	
Description	Gets the existing time values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.calendaring.Time[]	the existing time values
Errors	ILLEGAL_STATE	syntax is not a TIME or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getTypeSet**	
Description	Gets the set of acceptable Types.	
Return	osid.type.Type[]	a set of Types or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a TYPE
Compliance	mandatory	This method must be implemented.
Method	**getDefaultTypeValues**	

Description	Gets the default type values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.type.Type[]	the default type values
Errors	ILLEGAL_STATE	syntax is not a TYPE or isRequired() is true
Compliance	mandatory	This method must be implemented.
Method	**getExistingTypeValues**	
Description	Gets the existing type values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.type.Type[]	the existing type values
Errors	ILLEGAL_STATE	syntax is not a TYPE or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getVersionTypes**	
Description	Gets the set of acceptable version types.	
Return	osid.type.Type[]	the set of version types
Errors	ILLEGAL_STATE	syntax is not a VERSION
Compliance	mandatory	This method must be implemented.
Method	**supportsVersionType**	
Description	Tests if the given version type is supported.	
Parameters	osid.type.Type versionType	a version Type
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE	syntax is not a VERSION
	NULL_ARGUMENT	versionType is null
Compliance	mandatory	This method must be implemented.
Method	**getMinimumVersion**	
Description	Gets the minumim acceptable Version.	
Return	osid.installation.Version	the minumim Version
Errors	ILLEGAL_STATE	syntax is not a VERSION
Compliance	mandatory	This method must be implemented.
Method	**getMaximumVersion**	
Description	Gets the maximum acceptable Version.	
Return	osid.installation.Version	the maximum Version
Errors	ILLEGAL_STATE	syntax is not a VERSION
Compliance	mandatory	This method must be implemented.
Method	**getVersionSet**	
Description	Gets the set of acceptable Versions.	
Return	osid.installation.Version[]	a set of Versions or an empty array if not restricted
Errors	ILLEGAL_STATE	syntax is not a VERSION
Compliance	mandatory	This method must be implemented.
Method	**getDefaultVersionValues**	
Description	Gets the default version values. These are the values used if the element value is not provided or is cleared. If isArray() is false, then this method returns at most a single value.	
Return	osid.installation.Version[]	the default version values
Errors	ILLEGAL_STATE	syntax is not a TIME or isValueKnown() is false
Compliance	mandatory	This method must be implemented.
Method	**getExistingVersionValues**	
Description	Gets the existing version values. If hasValue() and isRequired() are false, then these values are the default values. If isArray() is false, then this method returns at most a single value.	
Return	osid.installation.Version[]	the existing version values

Errors	ILLEGAL_STATE	syntax is not a VERSION or isValueKnown() is false
Compliance	mandatory	This method must be implemented.

6.2. osid.OsidForm

Removes locale methods and uses DisplayText for string input (setJournalComment).

Interface	**osid.OsidForm**		
Implements			
Description	The Metadata interface defines a set of methods describing a the syntax and rules for creating and updating a data element inside an OsidForm. This interface provides a means to retrieve special restrictions placed upon data elements such as sizes and ranges that may vary from provider to provider or from object to object.		
Method	**isForUpdate**		
Description	Tests if this form is for an update operation.		
Return	boolean		true if this form is for an update operation, false if for a create operation
Compliance	mandatory		This method must be implemented.
Method	**getJournalCommentMetadata**		
Description	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.		
Return	osid.Metadata		a display label
Compliance	mandatory		This method must be implemented.
Method	**setJournalComment**		
Parameters	osid.locale.DisplayText	comment	the comment
Errors	INVALID_ARGUMENT		comment is invalid
	NO_ACCESS		Metadata.isReadonly() is true
	NULL_ARGUMENT		comment is null
Compliance	mandatory		This method must be implemented.
Method	**isValid**		
Description	Tests if this form is in a valid state for submission. A form is valid if all required data has been supplied compliant with any constraints.		
Return	boolean		false if there is a known error in this form, true otherwise
Errors	OPERATION_FAILED		attempt to perform validation failed
Compliance	mandatory		This method must be implemented.
Method	**getValidationMessages**		
Description	Gets text messages corresponding to additional instructions to pass form validation.		
Return	osid.locale.DisplayText[]		a list of messages
Compliance	mandatory		This method must be implemented.
Method	**getInvalidMetadata**		
Description	Gets a list of metadata for the elements in this form which are not valid.		
Return	osid.Metadata[]		invalid metadata
Compliance	mandatory		This method must be implemented.

7. Statement