

Existing & Default Values in Metadata

Status

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

Summary

OSID Consumers of create and update operations require knowledge of default values supplied by an OSID Provider that is currently not available via `OsidObjects` or `OsidForms`. This request describes enhancements to Metadata to make such information available.

Table of Contents

1. Current Specification.....	2
2. Problem	2
3. Proposed Change.....	3
3.1. Metadata Usage With Known Values	4
3.2. Metadata Usage With Unknown Values.....	5
3.3. Application Flow.....	7
4. Impacts	7
4.1. Specification	7
4.2. OSID Consumers.....	8
4.3. OSID Providers.....	8
5. Interoperability Considerations	8
6. Proposed Metadata Interface	9
7. Copyright Statement.....	23

1. Current Specification

For the purposes of this request, an `OsidObject` is any independently managed Identifiable with a corresponding `OsidForm`.

To create an `OsidObject`, an OSID Consumer requests an `OsidForm` for create. The `OsidForm` provides access to the elements that may be supplied by the OSID Consumer. The syntax of the element and its requiredness is communicated through the element's Metadata. For optional `OsidForm` elements that are mandatory in the corresponding `OsidObject`, it is presumed that the OSID Provider assigns some default value. For all elements, `Metadata.hasValue()` should be false since nothing has been previously supplied by an OSID Consumer.

To update an `OsidObject`, an OSID Consumer requests an `OsidForm` for update. The `OsidForm` provides access to the elements that may be changed. The syntax of the element and its ability to be modified is communicated via Metadata. `Metadata.hasValue()` may be true or false depending on if a default value has been assigned by an OSID Provider.

`Metadata.hasValue()` helps an OSID Consumer discern between what it is seeing in the `OsidForm` and the accompanying `OsidObject`. The information in the `OsidObject` may have been assigned by either the OSID Consumer or defaulted by the OSID Provider.

`OsidForms` typically have a mechanism to "clear" an element's value if the element's `Metadata.isRequired()` is false. The intent of these clear methods remove the value set by an OSID Consumer in a prior create or update operation. The OSID Provider must revert the element to some default value if it is also mandatory in the corresponding `OsidObject`.

2. Problem

1. An OSID Consumer must consume both the read-only `OsidObject` and the `OsidForm` if it wishes to communicate the existing data for updates. This requires a data-level orchestration between two categories of functionality where there is typically a service boundary. For example, what is seen through an `OsidObject` retrieved from a lookup or search session should line up with what the `OsidForm` retrieved from an admin session is communicating. In some scenarios, this alignment requires OSID Consumers to use different OSID Providers, one whose `OsidObjects` are tuned for user viewing and another tuned for administration.
2. An OSID Consumer has no way of knowing what these default values are until the create or update operation is processed.

3. Proposed Change

Expand the Metadata definition to include for each primitive, the existing data value and what the default value would be if not set or cleared. String example:

```
/**
 * 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 true if the element value is known, false if the element
 *         value is not known
 */
boolean isValueKnown();

/**
 * 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 at most returns a single value.
 *
 * @return the default string values
 * @throws ILLEGAL_STATE syntax is not a STRING or isRequired() is true
 */
String[] Metadata.getDefaultStringValue();

/**
 * 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 at most returns a single value.
 *
 * @return the existing string values
 * @throws ILLEGAL_STATE syntax is not a STRING or isValueKnown() is false
 */
String[] Metadata.getExistingStringValue();
```

Because an OsidForm input field may accept multiple values, the methods were defined as returning an array of primitives. If `Metadata.isArray()` is false, then a single value should be returned from these methods to be compliant.

The availability of default values is contingent upon the `Metadata.isRequired()`. If an element value is required, then the OSID Consumer must supply a value and the element value cannot be cleared on update.

The availability of the existing element values is contingent upon `Metadata.isValueKnown()`. `Metadata.hasValue()` is changed to also be contingent upon `Metadata.isValueKnown()`.

The following lists scenarios in the operation of Metadata with the proposed changes.

3.1. Metadata Usage With Known Values

Creation of Optional String Element With Known Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== true
Metadata.hasValue()	== false
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== {}

Creation of Required String Element With Known Value

Metadata.isRequired()	== true
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== true
Metadata.hasValue()	== false
Metadata.getDefaultStringValues()	== NA
Metadata.getExistingStringValues()	== {}

Creation of a Read-Only String Element With Known Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== true
Metadata.isValueKnown()	== true
Metadata.hasValue()	== false
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== {}

Update of Optional String Element With Known and Existing Default Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== true
Metadata.hasValue()	== false
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== {"default value"}

Update of Optional String Element With Known and Existing Non-Default Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== true
Metadata.hasValue()	== true
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== {"existing value"}

Update of Required String Element With Known Value

Metadata.isRequired()	== true
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== true
Metadata.hasValue()	== true
Metadata.getDefaultStringValues()	== NA
Metadata.getExistingStringValues()	== {"existing value"}

Update of a Read-Only String Element With Known Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== true
Metadata.isValueKnown()	== true
Metadata.hasValue()	== true
Metadata.getDefaultStringValues()	== {}
Metadata.getExistingStringValues()	== {"existing value"}

3.2. Metadata Usage With Unknown Values

Creation of Optional String Element With Unknown Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== false
Metadata.hasValue()	== NA
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== NA

Creation of Required String Element With Unknown Value

Metadata.isRequired()	== true
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== false
Metadata.hasValue()	== NA
Metadata.getDefaultStringValues()	== NA
Metadata.getExistingStringValues()	== NA

Creation of a Read-Only String Element With Unknown Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== true
Metadata.isValueKnown()	== false
Metadata.hasValue()	== NA
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== NA

Update of Optional String Element With With Unknown and Default Values

Metadata.isRequired()	== false
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== false
Metadata.hasValue()	== NA
Metadata.getDefaultStringValues()	== {"default value"}
Metadata.getExistingStringValues()	== NA

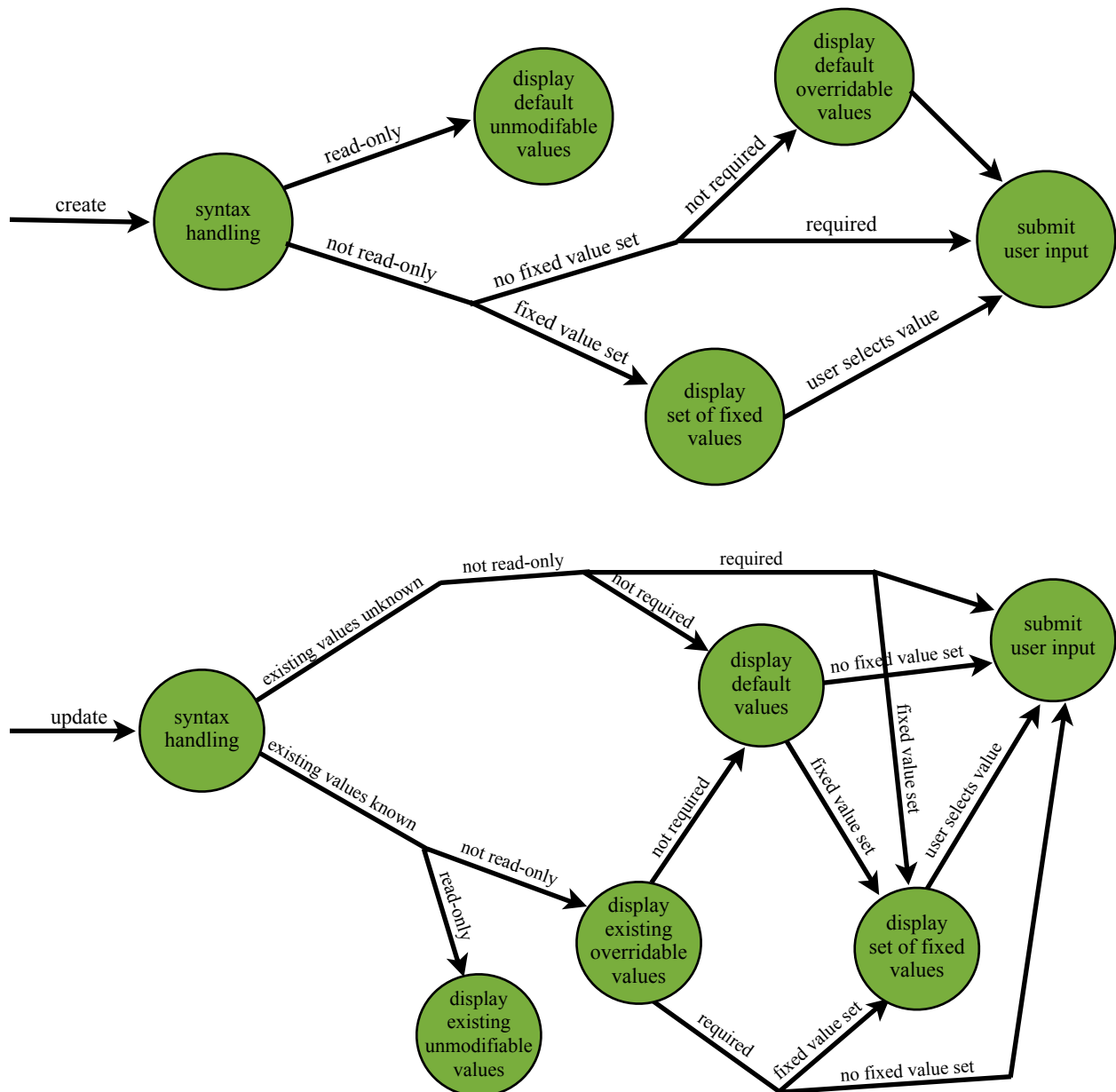
Update of Required String Element With Unknown Value

Metadata.isRequired()	== true
Metadata.isReadOnly()	== false
Metadata.isValueKnown()	== false
Metadata.hasValue()	== NA
Metadata.getDefaultStringValues()	== NA
Metadata.getExistingStringValues()	== NA

Update of a Read-Only String Element With Unknown Value

Metadata.isRequired()	== false
Metadata.isReadOnly()	== true
Metadata.isValueKnown()	== true
Metadata.hasValue()	== false
Metadata.getDefaultStringValues()	== {}
Metadata.getExistingStringValues()	== NA

3.3. Application Flow



4. Impacts

4.1. Specification

Changes are isolated to the Metadata interface used by all OsidForms. Changes are restricted to the addition of new methods. Metadata would be significantly larger in terms of overall number of methods, but the growing size is also due to having a single metadata

interface as opposed to a specific one for each primitive. The single interface model does help an OSID Consumer construct handlers for painting screens.

4.2. OSID Consumers

No impacts on OSID Consumers since no interface methods are changed or removed. However, the complexity around the interpretation of Metadata has increased for those OSID Consumers taking advantage of these new mechanisms.

4.3. OSID Providers

OSID Providers would be expected to implement the new Metadata methods. For create operations, OSID Providers would need to supply default values in Metadata. For update operations, OSID Providers would need to supply default values to communicate what occurs upon a clear operation. OSID Providers may optionally supply the existing values if `Metadata.isValueKnown()` and `Metadata.hasValue()` are both true.

5. Interoperability Considerations

This change to Metadata allows an OSID Consumer to understand existing element values within the context of update operations without having to infer them through an `OsidObject`. A single OSID Provider can support both retrieval and update of `OsidObjects` where the view through the `OsidObject` differs from how the input data is managed.

The OSID Provider can optionally provide more information about the underlying data allowing for both data aware and data blind providers.

There is no change in that an OSID Provider must have access to the existing `OsidObject Id` to support a single update operation. The existing specification also requires fulfillment of `Metadata.hasValue()`. OSID Providers were unable perform update operations without any knowledge of the underlying data because `Metadata.hasValue()` could not communicate "I don't know."

The proposed method, `Metadata.isValueKnown()`, communicates that the OSID Provider does not know about the existing element values. This splits the path for the OSID Consumer on update, but allows for updates blind to the underlying data outside the `Id` verification for single update operations.

6. Proposed Metadata Interface

<i>Interface</i>	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 NULL_ARGUMENT	syntax is not a CURRENCY 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 NULL_ARGUMENT	syntax is not a DATETIME or DURATION 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 NULL_ARGUMENT	syntax is not a DATETIME, DURATION, or TIME 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	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 NULL_ARGUMENT	syntax is not a HEADING 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 NULL_ARGUMENT UNSUPPORTED	syntax is not a HEADING headingType is null 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 NULL_ARGUMENT UNSUPPORTED	syntax is not a HEADING headingType is null 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 NULL_ARGUMENT	syntax is not a HEADING 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
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE NULL_ARGUMENT	syntax is not an OBJECT 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	spatialUnitRecordTyp
Return	boolean	a spatial unit record Type
Errors	ILLEGAL_STATE NULL_ARGUMENT	true if the type is supported, false otherwise syntax is not an SPATIALUNIT 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 restricted
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	
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	
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	
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	
	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	
	UNSUPPORTED	supportsStringMatchType(stringMatchType) is false	
Compliance	mandatory	This method must be implemented.	
Method	getStringFormatTypes		
Description	Gets the set of valid string formats.		
Return	osid.type.Type[]	the set of valid text format types	
Errors	ILLEGAL_STATE	syntax is not a STRING	
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	getDefaultStringValue		

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	getExistingStringValue	
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
Return	boolean	true if the type is supported, false otherwise
Errors	ILLEGAL_STATE NULL_ARGUMENT	syntax is not a VERSION versionType is null
Compliance	mandatory	This method must be implemented.
Method	getMinimumVersion	
Description	Gets the minimum acceptable Version.	
Return	osid.installation.Version	the minimum 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.

7. Copyright Statement

Copyright (C) Ingenescus (2013). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the authors, Ingenescus, or other organizations, except as required to translate it into languages other than English.

This document and the information contained herein is provided on an "AS IS" basis and Ingenescus and the authors DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.