Template version: proc-RFC_template-1_01_011016.dot

OSGi™

OPEN SERVICES GATEWAY INITIATIVE

RFC 21 − Measurement API

Confidential, Draft
rfc-0021-Measurement

24 Pages

Abstract

Data sources, such as electrical measuring instruments, speedometers and water tanks (pressure, level, etc) will generate state and measurement information. This RFC presents a simple and general API that provides a standardized encapsulation of measurements and states.

0 Document Information

0.1 Table of Contents

All Page Within This Box

0.2 Status

This document specifies a Measurement API for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

All Page Within This Box

0.3 Acknowledgement

This work began in VEG which did performed the lion share of the work while CPEG was busy completing SPR2.

0.4 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

```
Source code is shown in this typeface.
```

0.5 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	2001-06-15	First draft Erwin Morrhey, Jon-Erling Dahl
First CPEG draft	2001-10-16	Substantially modified RFC. Only org.osgi.util.measurement package remains. Cleaned up implementation and javadoc comments. The following areas were modified/clarified: • Exponent overflow/underflow • Time of new measurement • Comparable for Measurement • Add String name to State BJ Hargrave, IBM, hargrave@us.ibm.com
Second CPEG Draft	2001-10-24	Corrected implementation bugs. Replaced deg with rad. Exponents are 7 bits not 8 bits now. Removed copy constructors. Peter Kriens, OSGi , pkriens@aQute.se BJ Hargrave, IBM, hargrave@us.ibm.com

1 Introduction

These classes provide a standardized way to encapsulate measurements.

2 Motivation and Rationale

Data sources, such as electrical measuring instruments, speedometers and water tanks (pressure, level, etc) will generate state and measurement information. This RFC presents a simple and general API that provides a standardized encapsulation of measurements and states.

3 Technical Discussion

Package Class **Tree** **Deprecated** **Index** **Help**
PREV PACKAGE NEXT PACKAGE **FRAMES** **NO FRAMES**

3.1 Package org.osgi.util.measurement Description

The OSGi Measurement Package. Specification Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. For example:

```
Import-Package: org.osgi.util.measurement; specification-version=1.0
```

3.2 org.osgi.util.measurement
Class Measurement

```
java.lang.Object
```

```
  |
  +-org.osgi.util.measurement.Measurement
```

All Implemented Interfaces:
java.lang.Comparable

---

public class **Measurement**
extends java.lang.Object
implements java.lang.Comparable

Groups a value, error, unit and timestamp.

A Measurement is used for maintaining the tuple of value, error, unit and timestamp. The value and error are represented as doubles and the time is measured in milliseconds since midnight, January 1, 1970 UTC.

Mathemematic methods are provided that correctly calculate taking the error into account. A runtime error will occur when two measurements are used in an incompatible way. *e.g.* when a speed (m/s) is added to a distance (m). The measurement class will correctly track changes in unit during multiplication and division, always coercing the result to the most simple form. See `Unit` for more information on the supported units.

Errors in the measurement class are absolute errors. Measurement errors should use the P95 rule. Actual values must fall in the range value error 95% or more of the time.

A `Measurement` object is immutable in order to be easily shared.

Note: This class has a natural ordering that is inconsistent with equals. See `compareTo(java.lang.Object)`.

Version:
$Revision: 1.6 $
Author:
Open Services Gateway Initiative

Constructor Summary
`Measurement``(double value)` Create a new `Measurement` object with an error of 0.0, a unit of `Unit.unity` and a time of 0.
`Measurement``(double value, double error, Unit unit)` Create a new `Measurement` object with a time of 0.
`Measurement``(double value, double error, Unit unit, long time)` Create a new `Measurement` object.
`Measurement``(double value, Unit unit)` Create a new `Measurement` object with an error of 0.0 and a time of 0.

Method Summary	
`Measurement`	**`add`**`(double d)` Returns a new `Measurement` object that is the sum of this object added to the specified value.

`Measurement`	`add(double d, Unit u)` Returns a new `Measurement` object that is the sum of this object added to the specified value.
`Measurement`	`add(Measurement m)` Returns a new `Measurement` object that is the sum of this object added to the specified object.
`int`	`compareTo(java.lang.Object obj)` Compares this object with the specified object for order.
`Measurement`	`div(double d)` Returns a new `Measurement` object that is the quotient of this object divided by the specified value.
`Measurement`	`div(double d, Unit u)` Returns a new `Measurement` object that is the quotient of this object divided by the specified value.
`Measurement`	`div(Measurement m)` Returns a new `Measurement` object that is the quotient of this object divided by the specified object.
`boolean`	`equals(java.lang.Object obj)` Return whether the specified object is equal to this object.
`double`	`getError()` Returns the error of this `Measurement`.
`long`	`getTime()` Returns the time at which this `Measurement` was taken.
`Unit`	`getUnit()` Returns the `Unit` of this `Measurement`.
`double`	`getValue()` Returns the value of this `Measurement`.
`int`	`hashCode()` Returns a hash code value for this object.
`Measurement`	`mul(double d)` Returns a new `Measurement` object that is the product of this object multiplied by the specified value.
`Measurement`	`mul(double d, Unit u)` Returns a new `Measurement` object that is the product of this object multiplied by the specified value.
`Measurement`	`mul(Measurement m)` Returns a new `Measurement` object that is the product of this object multiplied by the specified object.
`Measurement`	`sub(double d)` Returns a new `Measurement` object that is the subtraction of the specified value from this object.
`Measurement`	`sub(double d, Unit u)` Returns a new `Measurement` object that is the subtraction of the specified value from this object.

`Measurement`	`sub(Measurement m)` Returns a new `Measurement` object that is the subtraction of the specified object from this object.
`java.lang.String`	`toString()` Returns a `String` representation of this `Measurement`

Methods inherited from class java.lang.Object
`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Constructor Detail

3.2.1 Measurement

```
public Measurement(double value,
                   double error,
                   Unit unit,
                   long time)
```

Create a new `Measurement` object.

Parameters:

`value` - The value of the `Measurement`.

`error` - The error of the `Measurement`.

`unit` - The `Unit` in which the `Measurement`. If this argument is `null`, then the unit will be set to `Unit.unity`.

`time` - The time measured in milliseconds since midnight, January 1, 1970 UTC.

3.2.2 Measurement

```
public Measurement(double value,
                   double error,
                   Unit unit)
```

Create a new `Measurement` object with a time of 0.

Parameters:

`value` - The value of the `Measurement`.

`error` - The error of the `Measurement`.

`unit` - The `Unit` in which the `Measurement`. If this argument is `null`, then the unit will be set to `Unit.unity`.

3.2.3 Measurement

```
public Measurement(double value,
                   Unit unit)
```

Create a new `Measurement` object with an error of 0.0 and a time of 0.

Parameters:

`value` - The value of the `Measurement`.

`unit` - The `Unit` in which the `Measurement`. If this argument is `null`, then the unit will be set to `Unit.unity`.

3.2.4 Measurement

`public **Measurement**(double value)`

Create a new `Measurement` object with an error of 0.0, a unit of `Unit.unity` and a time of 0.

Parameters:

`value` - The value of the `Measurement`.

Method Detail

3.2.5 getValue

`public final double **getValue**()`

Returns the value of this `Measurement`.

Returns:

The value of this `Measurement` as a double.

3.2.6 getError

`public final double **getError**()`

Returns the error of this `Measurement`. The error is always a positive value.

Returns:

The error of this `Measurement` as a double.

3.2.7 getUnit

`public final Unit **getUnit**()`

Returns the `Unit` of this `Measurement`.

Returns:

The `Unit` of this `Measurement`.

See Also:

`Unit`

3.2.8 getTime

`public final long **getTime**()`

Returns the time at which this `Measurement` was taken. The time is measured in milliseconds since midnight, January 1, 1970 UTC.

Returns:

the time at which this `Measurement` was taken.

3.2.9 mul

`public Measurement **mul**(Measurement m)`

Returns a new `Measurement` object that is the product of this object multiplied by the specified object.

Parameters:

`m` - The `Measurement` object that will be multiplied with this object.

Returns:

All Page Within This Box

A new Measurement that is the product of this object multiplied by the specified object. The error and Unit of the new object are computed. The time of the new object is set to the time of this object.

Throws:

ArithmeticException - If the Units of this object and the specified object cannot be multiplied.

See Also:

Unit

3.2.10 mul

```
public Measurement mul(double d,
                       Unit u)
```

Returns a new Measurement object that is the product of this object multiplied by the specified value.

Parameters:

d - The value that will be multiplied with this object.

u - The Unit of the specified value.

Returns:

A new Measurement that is the product of this object multiplied by the specified value. The error and Unit of the new object are computed. The time of the new object is set to the time of this object.

Throws:

ArithmeticException - If the Units of this object and the specified value cannot be multiplied.

See Also:

Unit

3.2.11 mul

```
public Measurement mul(double d)
```

Returns a new Measurement object that is the product of this object multiplied by the specified value.

Parameters:

d - The value that will be multiplied with this object.

Returns:

A new Measurement that is the product of this object multiplied by the specified value. The error of the new object is computed. The Unit and time of the new object is set to the Unit and time of this object.

3.2.12 div

```
public Measurement div(Measurement m)
```

Returns a new Measurement object that is the quotient of this object divided by the specified object.

Parameters:

m - The Measurement object that will be the divisor of this object.

Returns:

A new Measurement that is the quotient of this object dividied by the specified object. The error and Unit of the new object are computed. The time of the new object is set to the time of this object.

Throws:

ArithmeticException - If the Units of this object and the specified object cannot be divided.

See Also:

Unit

3.2.13 div

```
public Measurement div(double d,
                                Unit u)
```

Returns a new Measurement object that is the quotient of this object divided by the specified value.

Parameters:

d - The value that will be the divisor of this object.

u - The Unit of the specified value.

Returns:

A new Measurement that is the quotient of this object dividied by the specified value. The error and Unit of the new object are computed. The time of the new object is set to the time of this object.

Throws:

ArithmeticException - If the Units of this object and the specified object cannot be divided.

See Also:

Unit

3.2.14 div

```
public Measurement div(double d)
```

Returns a new Measurement object that is the quotient of this object divided by the specified value.

Parameters:

d - The value that will be the divisor of this object.

Returns:

A new Measurement that is the quotient of this object dividied by the specified value. The error of the new object is computed. The Unit and time of the new object is set to the Unit and time of this object.

3.2.15 add

```
public Measurement add(Measurement m)
```

Returns a new Measurement object that is the sum of this object added to the specified object.

Parameters:

m - The Measurement object that will be added with this object.

Returns:

A new Measurement that is the sum of this object added to the specified object. The error and Unit of the new object are computed. The time of the new object is set to the time of this object.

Throws:

ArithmeticException - If the Units of this object and the specified object cannot be added.

See Also:

Unit

3.2.16 add

```
public Measurement add(double d,
                                Unit u)
```

Returns a new Measurement object that is the sum of this object added to the specified value.

Parameters:

d - The value that will be added with this object.

u - The Unit of the specified value.

Returns:

A new Measurement that is the sum of this object added to the specified value. The Unit of the new object is computed. The error and time of the new object is set to the error and time of this object.

Throws:

ArithmeticException - If the Units of this object and the specified value cannot be added.

See Also:
Unit

3.2.17 add

```
public Measurement add(double d)
```

Returns a new Measurement object that is the sum of this object added to the specified value.
Parameters:
d - The value that will be added with this object.
Returns:
A new Measurement that is the sum of this object added to the specified value. The error, Unit and time of the new object is set to the error, Unit and time of this object.

3.2.18 sub

```
public Measurement sub(Measurement m)
```

Returns a new Measurement object that is the subtraction of the specified object from this object.
Parameters:
m - The Measurement object that will be subtracted from this object.
Returns:
A new Measurement that is the subtraction of the specified object from this object. The error and Unit of the new object are computed. The time of the new object is set to the time of this object.
Throws:
ArithmeticException - If the Units of this object and the specified object cannot be subtracted.
See Also:
Unit

3.2.19 sub

```
public Measurement sub(double d,
                       Unit u)
```

Returns a new Measurement object that is the subtraction of the specified value from this object.
Parameters:
d - The value that will be subtracted from this object.
u - The Unit of the specified value.
Returns:
A new Measurement that is the subtraction of the specified value from this object. The Unit of the new object is computed. The error and time of the new object is set to the error and time of this object.
Throws:
ArithmeticException - If the Units of this object and the specified object cannot be subtracted.
See Also:
Unit

3.2.20 sub

```
public Measurement sub(double d)
```

Returns a new Measurement object that is the subtraction of the specified value from this object.
Parameters:
d - The value that will be subtracted from this object.
Returns:

A new `Measurement` that is the subtraction of the specified value from this object. The error, `Unit` and time of the new object is set to the error, `Unit` and time of this object.

3.2.21 toString

```
public java.lang.String toString()
```

Returns a `String` representation of this `Measurement`

Overrides:

`toString` in class `java.lang.Object`

Returns:

a `String` representation of this `Measurement`

3.2.22 compareTo

```
public int compareTo(java.lang.Object obj)
```

Compares this object with the specified object for order. Returns a negative integer, zero, or a positive integer if this object is less than, equal to, or greater than the specified object.

Note: This class has a natural ordering that is inconsistent with equals. For this method, another `Measurement` object is considered equal if there is some `x` such that

```
  getValue()-getError() <= x <= getValue()□□□()
```

for both `Measurement` objects being compared.

Specified by:

`compareTo` in interface `java.lang.Comparable`

Parameters:

`obj` - The object to be compared.

Returns:

A negative integer, zero, or a positive integer if this object is less than, equal to, or greater than the specified object.

Throws:

`ClassCastException` - If the specified object is not of type `Measurement`.

`ArithmeticException` - If the `Unit` of the specified `Measurement` object is not equal to the `Unit` of this object.

3.2.23 hashCode

```
public int hashCode()
```

Returns a hash code value for this object.

Overrides:

`hashCode` in class `java.lang.Object`

Returns:

A hash code value for this object.

3.2.24 equals

```
public boolean equals(java.lang.Object obj)
```

Return whether the specified object is equal to this object. Two `Measurement` objects are equal if they have same value, error and `Unit`.

Note: This class has a natural ordering that is inconsistent with equals. See `compareTo(java.lang.Object)`.

Overrides:
`equals` in class `java.lang.Object`
Parameters:
`obj` - The object to compare with this object.
Returns:
`true` if this object is equal to the specified object; `false` otherwise.

3.3 org.osgi.util.measurement
Class State

```
java.lang.Object
  |
  +-org.osgi.util.measurement.State
```

public class **State**
extends java.lang.Object

Groups a state name, value and timestamp.

The state itself is represented as an integer and the time is measured in milliseconds since midnight, January 1, 1970 UTC.

A `State` object is immutable so that it may be easily shared.

Version:
$Revision: 1.5 $
Author:
Open Services Gateway Initiative

Constructor Summary
`State``(int value, java.lang.String name)` Create a new `State` object with a time of 0.
`State``(int value, java.lang.String name, long time)` Create a new `State` object.

Method Summary	
`boolean`	**`equals`**`(java.lang.Object obj)` Return whether the specified object is equal to this object.
`java.lang.String`	**`getName`**`()` Returns the name of this `State`.
`long`	**`getTime`**`()` Returns the time with which this `State` was created.

int	getValue() Returns the value of this State.
int	hashCode() Returns a hash code value for this object.
java.lang.String	toString() Returns a String representation of this object.

Methods inherited from class java.lang.Object
clone, finalize, getClass, notify, notifyAll, wait, wait, wait

Constructor Detail

3.3.1 State

```
public State(int value,
             java.lang.String name,
             long time)
```

Create a new State object.

Parameters:

value - The value of the state.

name - The name of the state.

time - The time measured in milliseconds since midnight, January 1, 1970 UTC.

3.3.2 State

```
public State(int value,
             java.lang.String name)
```

Create a new State object with a time of 0.

Parameters:

value - The value of the state.

name - The name of the state.

Method Detail

3.3.3 getValue

```
public final int getValue()
```

Returns the value of this State.

Returns:

The value of this State object.

3.3.4 getTime

```
public final long getTime()
```

Returns the time with which this State was created.

Returns:

The time with which this State was created. The time is measured in milliseconds since midnight, January 1, 1970 UTC.

3.3.5 getName

```
public final java.lang.String getName()
```

Returns the name of this State.

Returns:

The name of this State object.

3.3.6 toString

```
public java.lang.String toString()
```

Returns a String representation of this object.

Overrides:

toString in class java.lang.Object

Returns:

a String representation of this object.

3.3.7 hashCode

```
public int hashCode()
```

Returns a hash code value for this object.

Overrides:

hashCode in class java.lang.Object

Returns:

A hash code value for this object.

3.3.8 equals

```
public boolean equals(java.lang.Object obj)
```

Return whether the specified object is equal to this object. Two State objects are equal if they have same value and name.

Overrides:

equals in class java.lang.Object

Parameters:

obj - The object to compare with this object.

Returns:

true if this object is equal to the specified object; false otherwise.

3.4 org.osgi.util.measurement
Class Unit

```
java.lang.Object
  |
  +--org.osgi.util.measurement.Unit
```

public class **Unit**

extends java.lang.Object

A unit system for measurements. This class contains definitions of the most common SI units.

Units in this class are unique.

For a description of the units and SI system see: http://physics.nist.gov/cuu/Units/units.html

This class only support exponents for the base SI units in the range -64 to +63. Any operation which produces an exponent outside of this range will result in a `Unit` with undefined exponents.

Version:
$Revision: 1.7 $

Author:
Open Services Gateway Initiative

Field Summary

`static Unit`	**A** The electric current unit `ampere (A)`
`static Unit`	**Bq** The activity (of a radionuclide) unit `becquerel (Bq)`.
`static Unit`	**C** The electric charge unit `coulomb (C)`.
`static Unit`	**cd** The luminous intensity unit `candela (cd)`
`static Unit`	**F** The capacitance unit `farad (F)`.
`static Unit`	**Gy** The absorbed dose unit `gray (Gy)`.
`static Unit`	**Hz** The frequency unit `hertz (Hz)`.
`static Unit`	**J** The energy unit `joule (J)`.
`static Unit`	**K** The temperature unit `kelvin (K)`
`static Unit`	**kat** The catalytic activity unit `katal (kat)`.
`static Unit`	**kg** The mass unit `kilogram (kg)`
`static Unit`	**lx** The illuminance unit `lux (lx)`.
`static Unit`	**m** The length unit `meter (m)`
`static Unit`	**m_s** The speed unit `meter per second (m/s)`

static Unit	m_s2 The acceleration unit meter per second squared (m/s^2)
static Unit	m2 The area unit square meter (m^2)
static Unit	m3 The volume unit cubic meter (m^3)
static Unit	mol The amount of substance unit mole (mol)
static Unit	N The force unit newton (N).
static Unit	Ohm The electric resistance unit ohm.
static Unit	Pa The pressure unit pascal (Pa).
static Unit	rad The angle unit radians (rad)
static Unit	s The time unit second (s)
static Unit	S The electric conductance unit siemens (S).
static Unit	T The magnetic flux density unit tesla (T).
static Unit	unity Unspecified Unit
static Unit	V The electric potential difference unit volt (V).
static Unit	W The power unit watt (W).
static Unit	Wb The magnetic flux unit weber (Wb).

Method Summary

boolean	equals(java.lang.Object obj) Checks whether this Unit is equal to the specified Unit.
int	hashCode() Returns the hash code for this object.
java.lang.String	toString() Returns a String representation of the Unit

Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Field Detail

3.4.1 unity

`public static final Unit` **`unity`**

Unspecified Unit

3.4.2 m

`public static final Unit` **`m`**

The length unit `meter (m)`

3.4.3 s

`public static final Unit` **`s`**

The time unit `second (s)`

3.4.4 kg

`public static final Unit` **`kg`**

The mass unit `kilogram (kg)`

3.4.5 K

`public static final Unit` **`K`**

The temperature unit `kelvin (K)`

3.4.6 A

`public static final Unit` **`A`**

The electric current unit `ampere (A)`

3.4.7 mol

`public static final Unit` **`mol`**

The amount of substance unit `mole (mol)`

3.4.8 cd

`public static final Unit` **`cd`**

The luminous intensity unit `candela (cd)`

3.4.9 m_s

```
public static final Unit m_s
```

The speed unit meter per second (m/s)

3.4.10 m_s2

```
public static final Unit m_s2
```

The acceleration unit meter per second squared (m/s^2)

3.4.11 m2

```
public static final Unit m2
```

The area unit square meter (m^2)

3.4.12 m3

```
public static final Unit m3
```

The volume unit cubic meter (m^3)

3.4.13 Hz

```
public static final Unit Hz
```

The frequency unit hertz (Hz).

hertz is expressed in SI units as **1/s**

3.4.14 N

```
public static final Unit N
```

The force unit newton (N).

N is expressed in SI units as **m·kg/s^2**

3.4.15 Pa

```
public static final Unit Pa
```

The pressure unit pascal (Pa).

Pa is equal to **N/m^2** or is expressed in SI units as kg/m·s^2

3.4.16 J

```
public static final Unit J
```

The energy unit joule (J).

joule is equal to **N·m** or is expressed in SI units as **m^2·kg/s^2**

3.4.17 W

```
public static final Unit W
```

The power unit `watt (W)`.

watt is equal to **J/s** or is expressed in SI units as $\mathbf{m^2 \cdot kg/s^3}$

3.4.18 C

```
public static final Unit C
```

The electric charge unit `coulomb (C)`.

coulomb is expressed in SI units as $\mathbf{s \cdot A}$

3.4.19 V

```
public static final Unit V
```

The electric potential difference unit `volt (V)`.

volt is equal to **W/A** or is expressed in SI units as $\mathbf{m^2 \cdot kg/s^3 \cdot A}$

3.4.20 F

```
public static final Unit F
```

The capacitance unit `farad (F)`.

farad is equal to **C/V** or is expressed in SI units as $\mathbf{s^4 \cdot A^2/m^2 \cdot kg}$

3.4.21 Ohm

```
public static final Unit Ohm
```

The electric resistance unit `ohm`.

ohm is equal to **V/A** or is expressed in SI units as $\mathbf{m^2 \cdot kg/s^3 \cdot A^2}$

3.4.22 S

```
public static final Unit S
```

The electric conductance unit `siemens (S)`.

siemens is equal to **A/V** or is expressed in SI units as $\mathbf{s^3 \cdot A^2/m^2 \cdot kg}$

3.4.23 Wb

```
public static final Unit Wb
```

The magnetic flux unit `weber (Wb)`.

weber is equal to **V·s** or is expressed in SI units as **m^2·kg/s^2·A**

3.4.24 T

```
public static final Unit T
```

The magnetic flux density unit `tesla (T)`.

tesla is equal to **Wb/m^2** or is expressed in SI units as **kg/s^2·A**

3.4.25 lx

```
public static final Unit lx
```

The illuminance unit `lux (lx)`.

lux is expressed in SI units as **cd/m^2**

3.4.26 Gy

```
public static final Unit Gy
```

The absorbed dose unit `gray (Gy)`.

Gy is equal to **J/kg** or is expressed in SI units as **m^2·s^2**

3.4.27 kat

```
public static final Unit kat
```

The catalytic activity unit `katal (kat)`.

katal is expressed in SI units as **mol/s**

3.4.28 rad

```
public static final Unit rad
```

The angle unit `radians (rad)`

3.4.29 Bq

```
public static final Unit Bq
```

The activity (of a radionuclide) unit `becquerel (Bq)`.

Bq is expressed in SI units as **1/s**

Method Detail

3.4.30 equals

```
public boolean equals(java.lang.Object obj)
```

Checks whether this `Unit` is equal to the specified `Unit`. The `Units` are considered equal if their types are equal.

Overrides:

`equals` in class `java.lang.Object`

Parameters:

`o` - the `Unit` that should be checked for equality

Returns:

true if the specified `Unit` is equal to this `Unit`

3.4.31 hashCode

```
public int hashCode()
```

Returns the hash code for this object.

Overrides:

`hashCode` in class `java.lang.Object`

Returns:

This object's hash code.

3.4.32 toString

```
public java.lang.String toString()
```

Returns a `String` representation of the `Unit`

Overrides:

`toString` in class `java.lang.Object`

Returns:

a `String` representation of the `Unit`

4 Security Considerations

These immutable classes have no security concerns.

All Page Within This Box

5 Document Support

5.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

5.2 Author's Address

Name BJ Hargrave
Company IBM
Address 11400 Burnet Road, Austin, TX 78758 USA
Voice +1 521 838 9938
e-mail hargrave@us.ibm.com

Name Erwin Morrhey
Company Acunia
Address Vanden Tymplestraat 35
Voice +32 16 31 00 20
e-mail erwin.morrhey@acunia.com

Name Jon-Erling Dahl
Company Ericsson automotive e-services
Address Krokslätts fabriker 30, 431 37 Mölndal, Sweden
Voice +46 708 32 91 80
e-mail jon-erling.dahl@emw.ericsson.se

5.3 Acronyms and Abbreviations

5.4 End of Document