Report of the

Java 5 Language PSM for DDS

Revision 1.0 Task Force  
to the  
OMG Platform Technical Committee

May 19, 2014

Document Number: ptc/2013-12-17

Task Force Chair: Sumant Tambe (RTI)

Chartered: December 14, 2012 (Burlingame, CA USA)

Comments Due Date: December 6, 2013

Expiry Date: June 27, 2014

Template: omg/13-05-01

Table of Contents

Deliverables 1

Publication directions 1

Specification 1

Machine-consumable documents 1

IPR Mode 2

Summary of DDS-PSM-Java RTF Activities 2

Formation 2

Revision / Finalization Task Force Membership 2

Issue Disposition: 2

Voting Record: 4

Summary of Changes Made 5

Disposition: Resolved 6

OMG Issue No: 16096 7

Title: The current Java PSM makes an implicit assumptions on member ordering 7

OMG Issue No: 18420 9

Title: ModifiableInstanceHandle should be removed 9

OMG Issue No: 18422 10

Title: Bucket accessors shall be removed 10

OMG Issue No: 18423 11

Title: Operations with Collection as parameter should provide a "varargs" alternative 11

OMG Issue No: 18424 12

Title: Inconsistency in close() operations wrt. "throw java.io.IOException" declaration 12

OMG Issue No: 18425 13

Title: The PolicyFactory class misses operations for Presentation and TopicData creation 13

OMG Issue No: 18426 14

Title: In DomainParticipantFactoryQoS interface withPolicy() and withPolicies() have wrong parameters types 14

OMG Issue No: 18427 15

Title: Error in implementation of TypeSupport.newTypeSupport() 15

OMG Issue No: 18428 16

Title: The DataReaderQos interface misses a getReliability() operation 16

OMG Issue No: 18429 17

Title: Error in Javadoc of DomainParticipantFactory.createParticipant(domain, qos, listener, statuses) 17

OMG Issue No: 18430 18

Title: StatusCondition should be immutable and have fluent API 18

OMG Issue No: 18582 19

Title: The resolution of the FTF Issue 15966 was partially applied 19

OMG Issue No: 18652 20

Title: Missing "not" 20

Disposition: Closed. No Change. 21

OMG Issue No: 18421 22

Title: RequestOffered.requestedOfferedContract() semantic should be removed 22

OMG Issue No: 18432 23

Title: In WaitSet all waitForCondition() operations should return the triggered conditions 23

Disposition: Deferred 24

OMG Issue No: 18431 25

Title: The WaitSet class should be aligned to the ISO C++ IDDS PSM with regard to the dispatch() operation and associated functor. 25

# Deliverables

## Publication directions

The acronym chosen is "DDS-PSM-Java " and the version number is "1.1", the normative specification documents will be available under the URL http://www.omg.org/spec/DDS-PSM-Java/1.1/.

Chosen acronym: DDS-PSM-Java

Version number: 1.1

## Specification

Revised specification (clean): ptc/2014-xx-xx

Revised specification (change-bar): ptc/2014-xx-xx

Title: Java 5 Language PSM for DDS (DDS-PSM-Java)

Doc Number: ptc/2014-xx-xx

Description: Java 5 Language PSM for DDS

Status: Normative

URL: http://www.omg.org/spec/DDS-PSM-Java/1.1/2014-xx-xx

## Machine-consumable documents

Description: Java archive for DDS-PSM-Java interfaces

Doc Number: ptc/2014-xx-xx

Filename: omgdds.jar

Normative: Yes

Dependencies: omgdds\_src.zip

URL: http://www.omg.org/spec/DDS-PSM-Java/20140501/omgdds.jar

Description: Compressed file for DDS-PSM-Java interface sources

Doc Number: ptc/2014-xx-xx

Filename: omgdds\_src.zip

Normative: Yes

Dependencies: None

URL: http://www.omg.org/spec/DDS-PSM-Java/20140501/omgdds\_src.zip

Description: Source code diffs off of ptc/2012-12-04

Doc Number: ptc/2014-xx-xx

Filename: issue\_diffs.zip

Normative: No

Dependencies: None

URL: http://www.omg.org/spec/DDS-PSM-Java/20140501/issue\_diffs.zip

# IPR Mode

IPR mode of base specification: TBD

# Summary of DDS-PSM-Java RTF Activities

## Formation

* Chartered By: Platform TC
* Charter Date: December 14, 2012 (Burlingame, CA, USA)
* Comments Due Date: December 6, 2013
* Report Due Date: May 19, 2014

## Revision / Finalization Task Force Membership

|  |  |  |
| --- | --- | --- |
| **Member** | **Organization** | **Status** |
| Angelo Corsaro | PrismTech | Charter |
| Sumant Tambe | Real-Time Innovations (RTI) | Charter (chair) |
| Virginie Watine | Thales | Charter |

## Issue Disposition:

|  |  |  |
| --- | --- | --- |
| **Disposition** | **Number of Occurrences** | **Meaning of Disposition** |
| Resolved | 13 | The RTF agreed that there is a problem that needs fixing, and has proposed a resolution (which may or may not agree with any resolution the issue submitter proposed) |
| Deferred | 1 | The RTF agrees that there is a problem that needs fixing, but did not agree on a resolution and deferred its resolution to a future RTF. |
| Transferred | 0 | The RTF decided that the issue report relates to another specification, and recommends that it be transferred to the relevant RTF. |
| Closed, no change | 2 | The RTF decided that the issue report does not, in fact, identify a problem with this (or any other) OMG specification. |
| Closed, Out of Scope | 0 | The RTF decided that the issue report is an enhancement request, and therefore out of scope for this or any future FTF or RTF working on this major version of the specification. The RTF has closed the issue without making any specification changes, but RFP or RFC submission teams may like to consider these enhancement requests when proposing future new major versions of the specification. |
| Duplicate or merged | 0 | This issue is either an exact duplicate of another issue, or very closely related to another issue: see that issue for disposition. |

## Voting Record:

|  |  |  |
| --- | --- | --- |
| **Poll No.** | **Closing date** | **Issues included** |
| 1 | 5 May 2014 | 16096, 18420, 18421, 18422, 18423, 18424, 18425, 18426, 18427, 18428, 18429, 18430, 18431, 18432, 18582, 18652 |

|  |  |
| --- | --- |
| **Voter** | **Vote in poll 1** |
| Angelo Corsaro |  |
| Sumant Tambe | Yes to all |
| Virginie Watine |  |

## Summary of Changes Made

The DDS-PSM-Java RTF made changes that:

* Corrected features that impeded implementation of the specification
* Clarified ambiguous aspects of the specification, especially with respect to certain error-prone constructions
* Provided additional convenience for users

Here is the RTF's categorization of the resolutions applied to the specification according to their impact on the clarity and precision of the specification:

|  |  |  |
| --- | --- | --- |
| Extent of Change | Number of Issues | OMG Issue Numbers |
| **Critical/Urgent** - Fixed problems with normative parts of the specification which prevented implementation work | 1 | 16096 |
| **Significant** - Fixed problems with normative parts of the specification that raised concern about implementability | 12 | 18420, 18421, 18422, 18423, 18424, 18425, 18426, 18427, 18428, 18432, 18582, 18652 |
| **Minor** - Fixed minor problems with normative parts of the specification | 3 | 18429, 18430, 18431 |
| **Support Text** -Changes to descriptive, explanatory, or supporting material. | 0 |  |

Disposition: Resolved

OMG Issue No: 16096

Title: The current Java PSM makes an implicit assumptions on member ordering

Source:

PrismTech (Dr. Angelo Corsaro, PhD.)

**Nature:** Uncategorized Issue

**Severity:**

Summary:

The current Java PSM makes an implicit assumptions on member ordering when defining DDS Types in POJO that is not inherently guaranteed by the Java Language Specification. The possible resolution for this issue is to require @ID annotations when defining DDS Types using POJOs. In this case the ID would be used to uniquely defining the order to the attribute members.

Proposed Resolution:

To ensure portable ordering, each class member must be annotated with @DDSId annotation defined as follows

// Make this annotation accessible at runtime via reflection.

@Retention(RetentionPolicy.RUNTIME)

// This annotation can only be applied to class FILEDS.

@Target(ElementType.FIELD)

public @interface DDSId {

int value();

}

No two members in a Java class shall have the same value for the @DDSId annotation.

The fields in the DDS structured type shall be in the order defined using the @DDSId annotations. If one or more fields do not have the @DDSId annotation, the behavior is undefined.

This annotation shall be defined in org.omg.dds.type package in file Annotation.java as follows.

package org.omg.dds.core;

import java.lang.annotation.\*;

@Retention(RetentionPolicy.RUNTIME)

@Target(ElementType.FIELD)

public @interface DDSId {

int value();

}

When all the fields are annotated with unique integral value, the following algorithm provides a portable order of the fields.

import org.omg.dds.type.\*;

Field[] fields = anObject.getClass().getDeclaredFields();

java.util.Arrays.sort(fields, new Comparator<Field>(){

public int compare(Field f1, Field f2)

{

String a1 = f1.getAnnotation(DDSId.class).toString();

String a2 = f2.getAnnotation(DDSId.class).toString();

return a1.compareTo(a2);

}

});

If one or more fields do not have the @DDSId annotation, the behavior is undefined.

Proposed Disposition: Resolved

OMG Issue No: 18420

Title: ModifiableInstanceHandle should be removed

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Revision

**Severity:** Significant

Summary:

The ModifiableInstanceHandle was not removed along with all other modifiable types.

As such it should be removed for consistency, useability and performances.

Specificaly the following methods should be removed:

- ServiceEnvironment.newInstanceHandle()

- DataReader.lookupInstance(ModifiableInstanceHandle handle, TYPE keyHolder)

- DataWriter.lookupInstance(ModifiableInstanceHandle handle, TYPE keyHolder)

And the following method should be modified to return InstanceHandle:

- Sample.getInstanceHandle()

- Sample.getPublicationHandle()

Notice: the returning ModifialbeInstanceHandle forces defensive copying with obvious consequences in term of performances.

**Proposed Resolution:**

ModifiableInstanceHandle removed. datadistrib4j revision #220 and #222

**Proposed Disposition:** Resolved

OMG Issue No: 18422

Title: Bucket accessors shall be removed

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Enhancement

**Severity:** Significant

Summary:

Bucket accessors were supposed to be removed for all types (with exception of Time) as part of the FTF.

Apparently some bucket accessors are still present (in some cases as the only possibility - e.g. Subscriber.getDataReaders()).

This is very unfortunate since it forces defensive copies with the obvious performance impacts.

Furthermore it makes the API inconsistent.

**Proposed Resolution:**

Subscriber.getDataReaders() no longer has bucket accessors. However, bucket pattern is optionally available other places such as DataWrier.getKeyValue, DataReader.getKeyValue, etc. datadistrib4j revision #223

**Proposed Disposition: Resolved.**

OMG Issue No: 18423

Title: Operations with Collection as parameter should provide a "varargs" alternative

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Enhancement

**Severity:** Significant

Summary:

The use of Collection for providing a single "parameter" is rather cumbersome and could be greatly improved by providing a "varargs" alternative.

Here is an example that showcases the issue:

p = dp.createPublisher(

dp.getDefaultPublisherQos().withPolicy(

pf.Partition().withName(Arrays.asList("MyPartition"))

));

In this case a withName(String... names) operation will simplify the code as shown below:

p = dp.createPublisher(

dp.getDefaultPublisherQos().withPolicy(

pf.Partition().withName("MyPartition")

));

Moreover the use of Collection type could lead to compilation error as in the following case:

statusCondition.setEnabledStatuses(Arrays.toList(DataAvailableStatus.class));

**Proposed Resolution:**

Partition interface now has

public Partition withName(String name);

public Partition withName(String… names);

statusCondition interface now has

public void setEnabledStatuses(Class<? extends Status>... statuses);

Empty statuses argument list implies all statuses.

datadistrib4j revision #225 and revision #229

**Proposed Disposition: Resolved.**

OMG Issue No: 18424

Title: Inconsistency in close() operations wrt. "throw java.io.IOException" declaration

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Revision

**Severity:** Significant

Summary:

The Entity interface inherits from java.io.Closeable and overrides the close() operation to NOT throw java.io.IOException. But some inherited entities such as DataReader and DataWriter might have to throw IOException on closure as their implementation might use sockets.

On the other hand the Sample.Iterator interface also inherits from java.io.Closeable but its close() operation declare to throw IOException. There should not be any IOException in this case since the Samples are already received by the DaraReader

**Proposed Resolution:**

Entity.close has been updated to include java.io.IOException.

datadistrib4j revision #226

**Proposed Disposition: Resolved.**

OMG Issue No: 18425

Title: The PolicyFactory class misses operations for Presentation and TopicData creation

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Revision

**Severity:** Significant

Summary:

The PolicyFactory class provides operation to create all Policies but Presentation and TopicData.

Also, the Representation() operation should be renamed as DataRepresentation() to ensure consistency with the other factory operations (each is named accordingly to the type it returns).

**Proposed Resolution:**

Added Presentation and TopicData in PolicyFactory.

Renamed Representation to DataRepresentation

datadistrib4j revision #227

**Proposed Disposition: Resolved.**

OMG Issue No: 18426

Title: In DomainParticipantFactoryQoS interface withPolicy() and withPolicies() have wrong parameters types

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Revision

**Severity:** Significant

Summary:

In DomainParticipantFactoryQoS interface the following operation are defined:

public DomainParticipantFactoryQos withPolicy(

QosPolicy.ForDataWriter policy);

public DomainParticipantFactoryQos withPolicies(

QosPolicy.ForDataWriter... policy);

In both operations the correct parameter type should be QosPolicy.ForDomainParticipantFactory.

**Proposed Resolution:**

Changed QosPolicy.ForDataWriter to QosPolicy.ForDomainParticipantFactory

datadistrib4j revision #228

**Proposed Disposition: Resolved.**

OMG Issue No: 18427

Title: Error in implementation of TypeSupport.newTypeSupport()

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Revision

**Severity:** Significant

Summary:

In TypeSupport class the newTypeSupport() operation is implemented as following:

public static <TYPE> TypeSupport<TYPE> newTypeSupport(

Class<TYPE> type,

ServiceEnvironment env)

{

return newTypeSupport(type, type.getClass().getName(), env);

}

The correct implementation should be:

public static <TYPE> TypeSupport<TYPE> newTypeSupport(

Class<TYPE> type,

ServiceEnvironment env)

{

return newTypeSupport(type, type.getName(), env);

}

Since type.getClass().getName() always return "java.lang.Class".

**Proposed Resolution:**

Implementation changed to use type.getName().

datadistrib4j revision #230

**Proposed Disposition: Resolved.**

OMG Issue No: 18428

Title: The DataReaderQos interface misses a getReliability() operation

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Revision

**Severity:** Significant

Summary:

The DataReaderQos specifies accessors for all QosPolicy.ForDataReader policies but Reliability.

**Proposed Resolution:**

Added

Reliability DataReaderQos.getReliability();

datadistrib4j revision #231

**Proposed Disposition: Resolved.**

OMG Issue No: 18429

Title: Error in Javadoc of DomainParticipantFactory.createParticipant(domain, qos, listener, statuses)

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Clarification

**Severity:** Minor

Summary:

The Javadoc description of the DomainParticipantFactory.createParticipant(domain, qos, listener, statuses) operation is following:

"This operation creates a new DomainParticipant object having default QoS and no listener..."

It should be replaced with following:

"This operation creates a new DomainParticipant object with the specified QoS and the specified listener..."

**Proposed Resolution:**

Updated the documentation as proposed.

datadistrib4j revision #232

**Proposed Disposition: Resolved.**

OMG Issue No: 18430

Title: StatusCondition should be immutable and have fluent API

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Enhacement

**Severity:** Minor

Summary:

The StatusCondition should be immutable and follow the same pattern than QoS policies:

public interface StatusCondition<ENTITY extends Entity<?, ?>>

extends Condition {

/\* statuses accessor \*/

public Set<Class<? extends Status>> getEnabledStatuses();

/\* parent accessor \*/

public ENTITY getParent();

/\* Copy this StatusCondition and override the value of the statuses \*/

public StatusCondition<ENTITY> withEnabledStatuses(Class<? extends Status>... statuses);

}

This allow simpler code such as following:

waitset.attachCondition(

reader.getStatusCondition().withEnabledStatuses(DataAvailableStatus.class));

**Proposed Resolution:**

The proposed changes are made to the StatusCondition.

datadistrib4j revision #233 and #235

**Proposed Disposition: Resolved.**

OMG Issue No: 18582

Title: The resolution of the FTF Issue 15966 was partially applied

Source:

PrismTech (Dr. Angelo Corsaro, PhD.)

**Nature:** Revision

**Severity:** Significant

Summary:

The resolution for the ISSUE 15996 required to remove all the operations of the kind:

"public void setQos(String qosLibraryName, String qosProfileName);"

from all DDS entities since this operation pollutes the DDS API with a specific way of configuring QoS policies. The accepted resolution was to introduce the concept of a QoSProvider that could provide QoS settings by fetching them from a URI.

That said the type org.omg.dds.core.Entity still contains the operation:

public void setQos(String qosLibraryName, String qosProfileName);

This should be removed!

**Proposed Resolution:**

Removed the following operation.

public void setQos(String qosLibraryName, String qosProfileName);

datadistrib4j revision #234

**Proposed Disposition: Resolved.**

OMG Issue No: 18652

Title: Missing "not"

Source:

Unknown

**Nature:** Revision

**Severity:** Significant

Summary:

"many of which do support Java EE"

should be

"many of which do not support Java EE"

**Proposed Resolution:**

Modified specification section 6.2. Added “not”.

**Proposed Disposition: Resolved.**

Disposition: Closed. No Change.

OMG Issue No: 18421

Title: RequestOffered.requestedOfferedContract() semantic should be removed

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Enhancement

**Severity:** Significant

Summary:

Instead of providing the requestedOfferedContract() method for RxO comparison, the inherited Compare implementation should follow the RxO rules.

For instance: Deadline(2) > Deadline(5)

This change simplify the API and avoids the creation of temporary object when trying to evaluate RxO. Notice that, in this case, the natural ordering is still available via the Duration type.

**Proposed Resolution:**

The RequestOffered<T> interface inherits from the Comparable<T> interface, which includes the compareTo(T) function. The policy classes (e.g., Deadline) that implement RequestOffered<T> interface must also implement the Comparable<T> interface, which will allow natural comparison without creating a temporary object of type Comparable<T>.

The requestedOfferedContract() function is available to get hold of the comparator object when needed. Its use is not required for every comparison.

**Proposed Disposition:** Closed. No change.

OMG Issue No: 18432

Title: In WaitSet all waitForCondition() operations should return the triggered conditions

Source:

PrismTech (Mr. Julien Enoch)

**Nature:** Enhacement

**Severity:** Significant

Summary:

In WaitSet class, all waitForCondition() operation should return a Collection of the triggered conditions. The "activeConditions" parameter should be removed.

**Proposed Resolution:**

The WaitSet.waitForCondition operation uses the “bucket” pattern to avoid repeated allocation of a collection object. As this operation is expected to be called frequently, the overhead of allocation can be significant. The activeCondition parameter passed by the user avoid repeated allocation and stress on the GC.

**Proposed Disposition: Closed. No change**

Disposition: Deferred

OMG Issue No: 18431

Title: The WaitSet class should be aligned to the ISO C++ IDDS PSM with regard to the dispatch() operation and associated functor.

Source:

PrismTech (Dr. Angelo Corsaro, PhD., angelo.corsaro(at)prismtech.com)

**Nature:** Enhancement

**Severity:** Minor

Summary:

The WaitSet class should be aligned to the ISO C++ IDDS PSM with regard to the dispatch() operation and associated functor.

Proposed Resolution: Deferred, due to lack of sufficient discussions.

Proposed Disposition: Deferred