

**Report of the
DDS for (Light-Weight) CORBA Component Model
1.0
(DDS4CCM)
Finalization Task Force 2
to the
OMG Platform Technical Committee
21 May 2010**

Document Number: ptc/2010-05-06
Task Force Chair(s): Virginie Watine

Specification

Revised specification (clean): ptc/2010-05-07
Revised specification (change-bar): ptc/2010-05-08

Accompanying documents

Inventory:	ptc/2010-05-10	Non-normative
Machine-readable files	ptc/2010-05-09	Normative

Template: omg/09-06-01

Table of Contents

Disposition: Resolved.....	7
<i>OMG Issue No: 14836.....</i>	<i>7</i>
Title: IDL3+ keyword question/issue.....	7
Disposition: Resolved.....	8
<i>OMG Issue No: 14845.....</i>	<i>8</i>
Title: read_last.....	8
Disposition: Resolved.....	9
<i>OMG Issue No: 14846.....</i>	<i>9</i>
Title: Typo, 09-10-25, ExtendePortType.....	9
Disposition: Resolved.....	10
<i>OMG Issue No: 14847.....</i>	<i>10</i>
Title: getter text should be clearer about the behavior.....	10
Disposition: Resolved.....	11
<i>OMG Issue No: 15160.....</i>	<i>11</i>
Title: Type, DDS Listen instead of DDS_Listen.....	11
Disposition: Resolved.....	12
<i>OMG Issue No: 15173.....</i>	<i>12</i>
Title: InstanceHandleManager, local?.....	12
Disposition: Resolved.....	13
<i>OMG Issue No: 14612.....</i>	<i>13</i>
Title: mirrorport in CCMComponentPortKind.....	13
Disposition: Resolved.....	15
<i>OMG Issue No: 15172.....</i>	<i>15</i>
Title: Supporting Content Filtered Topics.....	15
Disposition: Resolved.....	19
<i>OMG Issue No: 15225.....</i>	<i>19</i>
Title: inout data parameters.....	19
Disposition: Deferred.....	20
<i>OMG Issue No: 14060.....</i>	<i>20</i>
Title: Section 8.4.4 talks about threading, but this section is really under specified.....	20
Disposition: Deferred.....	21
<i>OMG Issue No: 15238.....</i>	<i>21</i>
Title: Let the user be able to instantiate one dds connector.....	21
Disposition: Closed, no change.....	22
<i>OMG Issue No: 14830.....</i>	<i>22</i>
Title: csl::on_inconsistent_topic.....	22
Disposition: Closed, no change.....	23
<i>OMG Issue No: 15174.....</i>	<i>23</i>
Title: ConnectorStatusListener::on_unexpected_stat.....	23
Disposition: Closed, no change.....	24
<i>OMG Issue No: 14825.....</i>	<i>24</i>
Title: DDS_TopicBase::key_fields.....	24

Summary of DDS4CCM FTF Activities

Formation

- Chartered By: PTC
- On: 11 December 2009, Long Beach
- Comments Due Date: 19 March 2010
- Report Due Date: 2 July 2010

Revision / Finalization Task Force Membership

Member	Organization	Status
Virginie Watine	Thales	Charter - Chair
Gerardo Pardo-Castellote	RTI	Charter
Angelo Corsaro	Prismtech	Charter
Mark Hayman	Northrop Grumman	Charter
Johnny Willemsen	RemedyIT	Charter
Char Wales	Mitre	Charter

Issue Disposition:

Disposition	Number of Occurrences	Meaning of Disposition
Resolved	9	The RTF/FTF 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	2	The RTF/FTF agrees that there is a problem that needs fixing, but did not agree on a resolution and deferred its resolution to a future RTF/FTF.
Transferred	0	The RTF/FTF decided that the issue report relates to another specification, and recommends that it be transferred to the relevant RTF.
Closed, no change	3	The RTF/FTF 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/FTF 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/FTF 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	13 May 2010	14830, 14836, 14845, 14846, 14847, 15160, 15173, 15174
2	21 May 2010	14612, 14825, 15172, 15225

Voter	Vote in poll 1	Vote in poll 2
Char Wales	YES	YES
Mark Hayman	YES	YES
Angelo Corsaro	YES	YES
Gerardo Pardo	No vote	--
Johnny Willemsen	YES	YES
Virginie Watine	YES	YES

Summary of Changes Made

The DDS4CCM FTF2 made changes that:

- Enhance the DDS reading ports so that they can now support DDS content-filtered subscriptions
- Turn the out sequence parameters to inout, to allow for smarter memory management
- Clarify some wording and correct a few typos.

Here is the FTF's categorization of the 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	0	
Significant - Fixed problems with normative parts of the specification that raised concern about implementability	0	
Minor - Fixed minor problems with normative parts of the specification	5	14836, 14612, 15172, 15173, 15225
Support Text -Changes to descriptive, explanatory, or supporting material.	4	14845, 14846, 14847, 15160

Disposition: Resolved

OMG Issue No: 14836

Title: IDL3+ keyword question/issue

Source:

Vanderbilt University (Mr. Jeff Parsons, j.parsons@vanderbilt.edu)

Summary:

I just noticed, in section 7.2.2.1, that 'array' is allowed as a type designator for template parameters. So now it is also a keyword? it isn't on the list of new keywords in 7.3.6.

Resolution:

Remove 'array' from allowed type designator.

Rationale: Purpose was to avoid as much as possible new keywords. Expressing the fact that the parameter type must be of type array does not appear to be such a significant use case that a new keyword is justified.

Revised Text:

Page 11, line 3¹

Discard line : (array meaning all array types)

Disposition: **Resolved**

¹Page and line information are related to the new version of the specification with change bars.

Disposition: Resolved

OMG Issue No: 14845

Title: read_last

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

read_last returns the last sample of all instances. Any DDS error when reading the data will be reported by an InternalError exception. read_all returns all samples of all instances. Any DDS error when reading the data will be reported by an InternalError exception. But, what is a DDS error, is this != DDS_RETCODE_OK? What todo with DDS_RETCODE_NO_DATA, also then throw an exception or return an empty sequence?

Resolution:

Add a better explanation

Revised Text:

Page 35, lines 30 and 32,

- Add after the first sentence, the following one:
“In case of no data, the resulting data will be a void sequence”.
- In the sentence that follows, change: “Any DDS error” → “Any other DDS error”

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14846

Title: Typo, 09-10-25, ExtendePortType

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

In 09-10-25, page 3, line 17, it says ExtendePortType instead of ExtendedPortType

Resolution:

Correct the typo.

Revised Text:

Page 3, line 17,
Insert the missing 'd' in "ExtendedPortType"

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14847

Title: getter text should be clearer about the behavior

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Consider that timeout and max_delivered_data act as maximum boundaries (not minimums to be enforced). These just mean that you will never wait more than timeout nor receive more than max_delivered_data data. Applied to your precise questions:

- 1) should directly return with the 50 samples.
- 2) should return after receiving the first sample.

Resolution:

Add a better explanation

Revised Text:

Page 36, line 20

Change the sentence:

“Get operations are performed with the following parameters”

→

“Get operations are meant to provide information that has not been previously communicated to the participant. They may wait until fresh information is available and are performed with the following parameters:”

Page 36, line 30

Change the sentences

“get_many returns all the available samples in the limits set by the attribute max_delivered_data. In case there are more available samples, only the first max_delivered_data are returned.”

→

“get_many returns all the available samples within the limits set by the attribute max_delivered_data. In case there are too many available samples, only the first max_delivered_data ones are returned, the others remaining available for a subsequent call.”

Disposition:

Resolved

Disposition: Resolved

OMG Issue No: 15160

Title: Type, DDS Listen instead of DDS_Listen

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

page 40 line one has a type, DDS Listen should be DDS_Listen for the port name

Resolution:

Correct the typo

Revised Text:

Page 40, line 39:
Change "DDS Listen" → "DDS_Listen"

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 15173

Title: InstanceHandleManager, local?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The InstanceHandleManager interface is defined as abstract, but shouldn't this one also be defined as local (which means remove abstract, you can't have both). By now having abstract, it is technically possible to derive a regular (non local) interface from it. This means we have todo more code generation as necessary.

We would like to propose to replace abstract with local.

Resolution:

Make this (abstract) interface local.

Revised Text:

Page 31, line 11

Change `abstract` → `local`

Page 31, line 17

Change "This interface" → "this abstract interface"

Page 54, line 47

Change `abstract` → `local`

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14612

Title: mirrorport in CCMComponentPortKind

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

In CCMComponentPortKind ExtendedPort and Mirrorport are added. First in update 5, page 11, line 21, font of MirrorPort is not ok. The other issue is that MirrorPort is a keyword, so it should be _MirrorPort or maybe ExtendedMirrorPort to match the ExtendedPort.

Resolution:

In addition to the change “MirrorPort” to “ExtendedMirrorPort”, correction of two other related mistakes

- “CCMPortKind” does not exist (“CCMComponentPortKind” instead)
- The name of the attribute is "kind" (“CCMComponentPortKind” is the name of the enumeration)

Revised Text:

Page 3, line 23

Change “MirrorPort” → “ExtendedMirrorPort”

Page 23, line 18

Change “CCMPortKind” → “CCMComponentPortKind”

Page 23, line 19

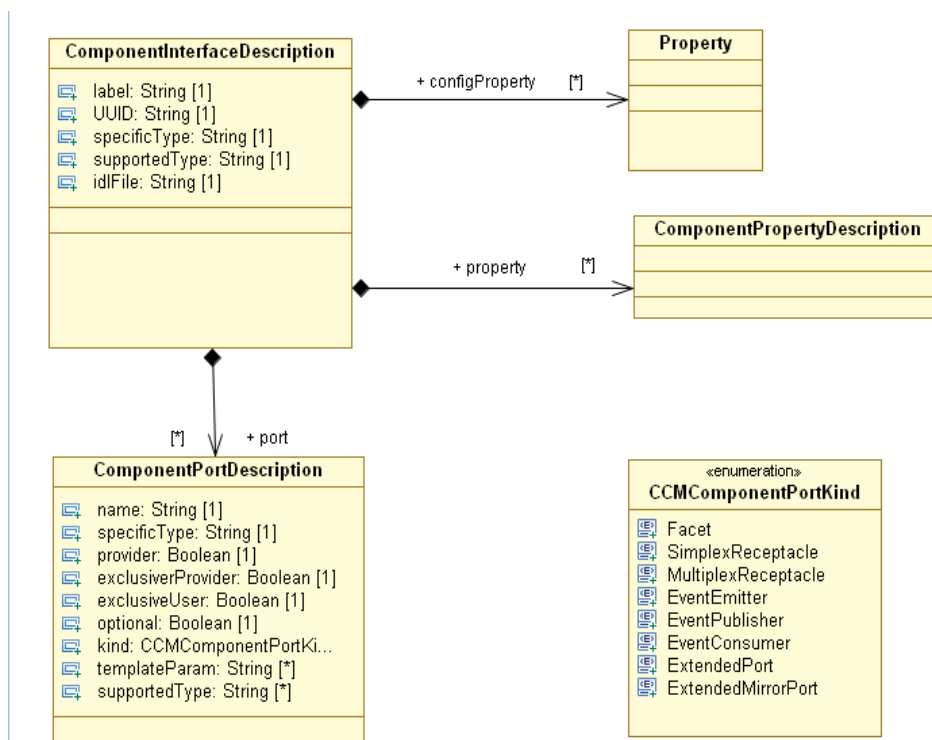
Change “MirrorPort” → “ExtendedMirrorPort”

Page 23, line 23

Change “CCMComponentPortKind” → “kind”

Change “MirrorPort” → “ExtendedMirrorPort”

Change Figure 9 Page 24 with the following:



Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 15172

Title: Supporting Content Filtered Topics

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The current QueryFilter in the spec seems to match best to a QueryCondition. But DDS also provides the ContentFilteredTopic support, that doesn't seem to be available at the ccm-dds level, is this intended?

Resolution:

The resolution is to add on the reading-side DDS ports a filter attribute and a basic port to change filter parameters when appropriate (this basic port will be used in the port and provided in the associated mirrorport). This disposal implies to add configuration attributes in porttype definitions.

In addition, for clarity purpose, the formerly named "filter" attribute on the Reader interface is renamed "query" and inside the QueryFilter structure, the first field (formerly "query") is renamed "expression" and the second (formerly "query_parameters") simplified to become "parameters".

Rationale:

- The current QueryFilter is definitively intended to match the DDS query. ContentFilteredTopic is also a very important DDS feature that should be accessible through ccm-dds.
- Configuration attributes in porttype definitions is also useful for many other uses of the GIS (was missing)

Revised Text:

Page 6, line 6

Add at the end of the sentence: "and attributes to configure them if needed."

Page 6, line 9

Add at the end of the sentence "and attributes."

Page 7, line 3

Add a semi-colon at the end of the line

Page 7, line 5

Change the ending dot into a semi-colon

Add two extra bullets

- An attribute in a port or a mirrorport becomes an attribute in the equivalent IDL3 declaration of the component;
- The name of the generated attribute is the concatenation of the extended port name and the related attribute name of the porttype, separated by '_'.

Page 13, line 19

Add | <attr_dcl> ";"

Page 15, line 22

Add | <attr_dcl> ";"

Page 15, line 26

Insert after "basic ports", "and attributes,"

Page 34, line 18

Change `query` → `expression`

Page 34, line 19

Change `query_parameters` → `parameters`

Page 35 , line 1, 4

Change "query" → "query expression"

Page 35, line 3

Change "query" → "expression"

Page 35, line 15

Change `filter` → `query`

page 35, line 29

Replace the line with: "Through the query as specified in the query ("" as expression means no query)"

Page 38, line 35

Insert a new section 8.1.2.2.4

8.1.2.2.4 Content Filter Management

In addition to plain topics, DDS provides content-filtered topics for content-based subscriptions. Such a topic has to be created in relation with a classical one and given a filter expression. All data provided by this topic must pass the filter expression. Apart that characteristic, content-filter topics and classical ones can be used the same way.

The following attribute allows declaring a filter to the port that will be used for DDS content-filtered subscriptions, in case it is given a value at configuration time.

Attribute Filter

```
attribute QueryFilter          filter
      setraises (NonChangeable);
```

While the filter expression is immutable and can be thus considered as a structural configuration attribute of a given port, its parameters can be modified dynamically.

The following interface allows changing those parameters.

Interface ContentFilterSetting

```
local interface ContentFilterSetting {
    void set_filter_parameters (in DDS::StringSeq parameters)
        raises (InternalError);
};
```

(note that previous section 8.1.2.2.4 is now 8.1.2.2.5)

Page 40, lines 22, 33, 43

Page 41, line 5

Insert the following

```
attribute QueryFilter          filter
      setraises(NonChangeable);
uses ContentFilterSetting      filter_config;
```

Page 51, line 15

Change the last sentence of the paragraph → “All the ports intended for the subscribing side comprise also a configuration attribute (filter) to set the content filter, a basic port to change the parameters of the filter expression (filter_config) and a port to be notified of the relevant statuses (status).”

Page 52, line 41

Change query → expression

Page 52, line 42

Change query_parameters → parameters

Page 54, line 31

Insert the following

```
// Content Filter Parameters Setting
// -----
local interface ContentFilterSetting {
    void set_filter_parameters (in DDS::StringSeq parameters)
```

```
        raises (InternalError);  
    };
```

Page 55, line 6

Change `filter` → `query`

page 56, line 7

Replace the line with:

```
//      - through the query as specified (" expression means no query)
```

Page 57, lines 40, 49

Page 58, lines 1, 12

Insert the following:

```
    attribute QueryFilter          filter  
        setraises(NonChangeable);  
    uses ContentFilterSetting      filter_config;
```

Disposition:

Resolved

Disposition: Resolved

OMG Issue No: 15225

Title: inout data parameters

Source:

THALES (Ms. Virginie Watine, virginie.watine@thalesgroup.com)

Summary:

Currently in the spec when several data are to be returned, the related parameter is an 'out' sequence. Even if this is semantically correct, it makes implementation of smart memory management impossible.

Suggestion is then to turn those parameters into 'inout' ones.

Resolution:

Change the sequence out parameters into inout.

Revised Text:

Page 30, line 13

Insert the following:

“Sequences to be returned (of data and of accompanying information) are designed as 'inout' parameters, even if the actual information flow is only 'out'. This disposal allows for implementation of smarter memory management.”

Page 35, line 9, 11, 17

Page 36, line 14

Page 56, lines 56, 58

Page 57, lines 2, 30

Change `out Tseq` → `inout Tseq`

Change `out readInfoSeq` → `inout ReadInfoSeq`

Disposition: **Resolved**

Disposition: Deferred

OMG Issue No: 14060

Title: Section 8.4.4 talks about threading, but this section is really under specified

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Section 8.4.4 talks about threading, but this section is really under specified. It should be much clearer how threading works and what guarantees are given.

Discussion:

No time to tackle it completely. In any case, will not change the already defined interfaces.

Disposition: **Deferred**

Disposition: Deferred

OMG Issue No: 15238

Title: Let the user be able to instantiate one dds connector

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Currently the DDS4CCM specification puts within the Typed module the extended ports, but also the dds connectors. The drawback of this approach is that when the user instantiates this templated module he gets both connectors, even when he is only interested in using one. We propose to move the DDS_State and DDS_Event to their own typed module so that the end user can instantiate just one of these connectors. This also makes it easier for a code generator to just generate the implementation for a specific connector, at this moment there is nothing in idl which can be used to determine whether the end user wants to use DDS_Event and/or DDS_State.

Discussion:

No time to tackle it completely. Other solutions to minimize code generation are under investigation.

Disposition: **Deferred**

Disposition: Closed, no change

OMG Issue No: 14830

Title: `cs::on_inconsistent_topic`

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

When we check the ConnectorStatusListener we see that it has an `on_inconsistent_topic`. The CSL is part of the DDS_Base connector, but the concept of a topic is added to the DDS_TopicBase which is derived from DDS_Base. It looks inconsistent that the CSL has a callback for something that is not known at that connector level.

Why do we have a DDS_Base and a DDS_TopicBase, why not merge them into one connector base?

Discussion:

DDS connectors not concerned with any topic make no sense but some connectors are concerned with several topics (and in particular DLRL's ones if we talk about standardized ones). If the root of all connectors were DDS_TopicBase (as suggested), all the attached topics would not be treated equally, which is quite strange.

- DDS_Base is the root for all connectors
- DDS_TopicBase:DDS_Base is the root for all mono-topic connectors

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 15174

Title: ConnectorStatusListener::on_unexpected_stat

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

On the ConnectorStatusListener there is an on_unexpected_status method. What is meant with an unexpected status, I don't find the spec clear in this area. If this are the statuses we don't handle explicitly, what can the user do with it? He gets the entity and the kind, but now the status field itself? Is this useful to add?

Discussion:

Unfortunately, it is not possible to get simply the status fields for unknown statuses... Fortunately, this should never happen (except if the DDS implementation bugs)

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14825

Title: DDS_TopicBase::key_fields

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

DDS_TopicBase::key_fields is read/write with nonchangeable. I am aware that DDS doesn't have a standardized way to specify keys in IDL. At the CCM level we always instantiate a connector for a specific IDL type. It looks a nice feature to be able to ask to the connector what the key_fields are, that is something the tooling can't get from the IDL itself. But, why should this be a writable (but not changeable) attribute, couldn't it just be readonly, it is just there for the component to retrieve, it can probably be implemented with some vendor specific code.

Discussion:

Having the key fields expressed in the CDP is a nice feature at least for system-wide documentation purpose.

If the DDS implementation does not provide any means to retrieve this information, the tooling can fill the gap by propagating it.

Disposition: **Closed, no change**