Report of the

DDS for (Light-Weight) CORBA Component Model 1.0 (DDS4CCM)

Finalization Task Force

to the

OMG Platform Technical Committee

9 November 2009

Document Number:	**ptc/2009-10-23**
Task Force Chair(s):	**Virginie Watine**

Specification

Revised specification (clean):	**ptc/2009-10-25**
Revised specification (change-bar):	**ptc/2009-10-26**

Accompanying documents

Inventory:	**ptc/2009-10-27**	**Non-normative**
Machine-readable files	**ptc/2009-10-24**	**Non-normative)**

Template: omg/09-06-01

Table of Contents

Summary of DDS4CCM FTF Activities

Formation

- Chartered By: PTC
- On: 12 December 2008, Santa Clara
- Comments Due Date: 31 August 2009
- Report Due Date: 18 December 2009

Revision / Finalization Task Force Membership

Member	Organization	Status
Virginie Watine	Thales	Charter - Chair
Gerardo Pardo-Castellote	RTI	Charter
Angelo Corsaro	Prismtech	Charter
Jim Kulp	Mercury Systems	Charter
Johnny Willemsen	RemedyIT	Charter
Char Wales	Mitre	Charter

Issue Disposition:

Disposition	Number of Occurrences	Meaning of Disposition
Resolved	53	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	8	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	37	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	1 September 2009	13891 13892 13896 13946 13953 13957 13959 13964 13965 13966 13969 13970 13971 14025 14080
2	1 September 2009	13954 13955 13956 13958 13961 13972
3	1 September 2009	13886 13888 13889 13890 13893 13897 13949 13952 13962 13963 13968 13973 14110 14017
4	9 November 2009	13960 14174 14175 14178 14179 14206 14207 14211
5	9 November 2009	13885 14024 14028 14037 14038 14039 14040 14119 14120 14121 14122 14201 13884 14020 14046 14057 14558 13894 13895
6	9 November 2009	14181 14182 14202 14208 14209 14210 14212 14213 14214 14534 14562 14567 14574 14578 14590 13974 13967 14168 14217 14581 14611 14060

Voter	Vote in poll 1	Vote in poll 2	Vote in poll 3
Char Wales	YES	YES	YES
Jim Kulp	YES Abstain (14080)	No Vote	No Vote
Angelo Corsaro	YES	YES	YES
Gerardo Pardo	YES	YES	YES
Johnny Willemsen	YES	YES	YES
Virginie Watine	YES	YES	YES

Voter	Vote in poll 4	Vote in poll 5	Vote in poll 6
Char Wales	YES	Abstain	Abstain
Jim Kulp	No Vote	Abstain	Abstain
Angelo Corsaro	YES	YES	YES
Gerardo Pardo	YES	YES	YES
Johnny Willemsen	YES	YES	YES NO (14202, 14590)
Virginie Watine	YES	YES	YES

Summary of Changes Made

The DDS4CCM FTF made changes that:

- Redesign the template support, part of the Generic Interaction Support (chapter 7), in order to better align it with IDL recent evolutions – notably the deprecation of anonymous types. Templates are now set at module level, which is less intrusive in the grammar while more generic, thus offering more use possibilities. This change leads to a total reorganization of chapter 7.
  The IDL3+ grammar has been rewritten accordingly and better aligned with the standard IDL grammar.

- Change the DDS-DCPS ports definition to take into account the new template support and some enhancements suggested by the FTF. This change has impacted the whole section 8.2.2 and the Annex A.

- Clarify some wording and correct many 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	**4**	**13884 14201 14213 14214**
Minor - Fixed minor problems with normative parts of the specification	**14**	**13888 13890 13854 13968 13973 14017 14117 14168 14177 14209 14567 14574 14590 14611**
Support Text -Changes to descriptive, explanatory, or supporting material.	**35**	**13886 13891 13892 13893 13896 13946 13953 13954 13955 13956 13957 13958 13959 13960 13961 13963 13964 13965 13966 13967 13970 13971 13972 13974 14024 14025 14174 14175 14176 14178 14179 14206 14207 14210 14212**

Disposition: Resolved

OMG Issue No: 13884

Title: push operation should be push_T$

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 7 / lines 22-26[1]
lists: // Parameterized interface interface EventsPusher <typename T> { typedef sequence<T> T$Seq; // construction of a type name void push (in T$Seq events); void push_$T (in T event); // construction of an operation name };
The replacement, is that T$ or $T, seems the push operation should be push_T$

Resolution:

The new template support in response to issue 14201 removes the need for that controversial concatenation operator.

Revised Text:

None as already covered by resolution of issue 14201

Disposition: **Resolved**

[1] If the issue description part, page and line numbers are refer to beta1 document, while is the revised text, they refer to the convenience document without change bars.

Disposition: Resolved

OMG Issue No: 13886

Title: Should be as below (lower case) 7.2.3.4.1 set_configuration_values

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 27;19 / line 18
Line 18, has 7.2.3.4.1 Set_configuration_values Should be as below (lower case) 7.2.3.4.1 set_configuration_values

Resolution:

Correct the typo

Revised Text:

Page 20 / line 12
Change Set_configuration_values → set_configuration_values

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13888

Title: wouldn't it be more maintainable to add to lines 35-40 a typedef for the return value (and other interfaces)?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 29 / lines 35-40
wouldn't it be more maintainable to add to lines 35-40 a typedef for the return value (and other interfaces)?
typedef unsigned long bytes;
Also change the comment to // returns number of bytes written
interface MultiWriter<typename T> { // T assumed to be a data type

```
        typedef sequence<T> T$Seq;
        nb_bytes write(in T$Seq instances) // returns nb of written
        raises (InternalError);
        attribute boolean is_coherent_write;
        };
```

Resolution:

The idea is good but the proposed name (bytes) is misleading as what is returned is a count of data samples. In addition, it in inconsistent to let unsigned long as index in lists of data. It is thus proposed to name that typedef DataNumber_t, to create also DataNumberSeq and to use them each time the unsigned long or sequence of Ulong were used.

Revised Text:

Page 29 / line 37
Change: `unsigned long` → `DataNumber_t`
Change in comment: `nb` → `number`

Page 30 / lines 29, 33, 37
Change: `unsigned long` → `DataNumber_t`
Change in comment: `nb` → `number`

Page 31 / line 34
Change: `unsigned long` → `DataNumber_t`

Page 48, after line 5,
Insert:

```
// --------
// Typedefs
// --------
typedef unsigned long         DataNumber_t
typedef sequence<DataNumber_t>      DataNumberSeq
```

Page 48 / lines 17, 21
Change: `ULlongSeq` → `DataNumberSeq`

Page 48 / lines 27
Change: `unsigned long` → `DataNumber_t`

Page 48 / lines 46
Change: `unsigned long` → `DataNumber_t`
Change in comment: `nb` → `number`

Page 48 / lines 23, 26, 29
Change `unsigned long` → `DataNumber_t`
Change in comment: `nb` → `number`

Page 50 / line 6
Change: `unsigned long` → `DataNumber_t`

However, further issue resolution (cf. Issue 14201) deprecates some of those changes.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13890

Title: Change on line 5 StringSeq to CORBAStringSeq

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 32 / line 5
Change StringSeq to CORBA::StringSeq

Resolution:

As CCM is meant to abstract from CORBA, it is not a good thing to create here a dependency to CORBA In addition, those StringSeq will be passed to DDS, DDS::StringSeq is therefore a much better choice.

This concerns the query parameters as well as the topic key fields.

Revised Text:

Page 34 / line 17
Page 41 / line 14
Page 51 / line 32
Change: `StringSeq` → `DDS::StringSeq`

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13891

Title: line 29, change On_data to on_data (lower case)

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 34 / line 29
change On_data to on_data (lower case),

Resolution:

Correct the typo

Revised Text:

Page 34 / line 29
Change: On_data → on_data,

However, further issue resolution (cf. Issue 14214) deprecates that change.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13892

Title: line 52, change DDS_StateLsisten to DDS_StateListen

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 36 / line 52,
change DDS_StateLsisten to DDS_StateListen

Resolution:

Correct the typo

Revised Text:

Page 36 / line 52,
Change: DDS_StateLsisten → DDS_StateListen

However, further issue resolution (cf. Issue 14214) deprecates that change.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13893

Title: line 27, change attributre to attribute also change StringSeq to CORBAStringSeq

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 37 / line 27
change attributre to attribute also change StringSeq to CORBA::StringSeq

Resolution:

Correct the typo (the second part of the issue is already in issue 13890)

Revised Text:

Page 41 / line 14
Change: `attributre` → `attribute`

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13894

Title: line 8, shouldn't DDS_KEEP_ALL_HISTORY_QOS be replaces with KEEP_ALL_HISTORY_QOS

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

line 8, shouldn't DDS_KEEP_ALL_HISTORY_QOS be replaces with KEEP_ALL_HISTORY_QOS

Resolution:

Discard DDS_ in front of all QoS values

Revised Text:

Page 43 / line 17
Page 46 / lines 6, 9, 14, 17, 35, 38, 43, 46
Page 47 / line 13
Remove: DDS_

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13896

Title: line 4, replace the with The (T upper case)

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 45 / line 4
replace the with The (T upper case)

Resolution:

Correct the typo

Revised Text:

Page 48 / line 4
Change: the → The

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13946

Title: Figure 1 should say IDL3+ instead of Extended IDL3

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 9
Figure 1 should say IDL3+ instead of Extended IDL3

Resolution:

Replace Extended IDL3 by IDL3+

Revised Text:

Page 5
Change: Extended IDL3 → IDL3+ in Figure 1
Change: IDL3 → IDL3+ in the title of Figure 1

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13953

Title: DDS_Listen is never defined in this spec

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 38 / line 19, Page 54 / line 12
The spec uses DDS_Listen<> on page 38 and 54, but DDS_Listen is never defined in this spec

Resolution:

The correct name is DDS_RawListen.

Revised Text:

Page 38 / line 19
Change: DDS_Listen → DDS_RawListen

Page 54 / line 12
Change: DDS_Listen → DDS_RawListen

However, further issue resolution (cf. Issue 14214) deprecates that change.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13954

Title: For 3, check UML CCM reference, it says 5formal, is that ok?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 2 / line 6
For 3, check UML CCM reference, it says 5formal, is that ok?

Resolution:

Correct the typo

Revised Text:

Page 2 / line 6

Change: 5 → (

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13955

Title: Section 6.1.2 It should be ExtendedPort and MirrorPort, not InversePort

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 3 / line 23
It should be ExtendedPort and MirrorPort, not InversePort

Resolution:

Correct the typo

Revised Text:

Page 3 / line 23
Change: InversePort → MirrorPort

Disposition: Resolved

Disposition: Resolved

OMG Issue No: 13956

Title: Section 7.1.2 Layout of figure 2 and 3 can be improved

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Layout of figure 2 and 3 can be improved, not all lines hit the connector, text is not all readable

Resolution:

Change the figures

Revised Text:

Page 7
Figures 2 and 3 changed (cf. Convenience document for the figures)

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13957

Title: Section 7.2.1.2 Line 16 ends with two semi colons instead of 1

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 15 / line 16
ends with two semi colons instead of 1

Resolution:

Correct the typo

Revised Text:

Page 23;15 / line 16
Discard extra semi-colon

However, further issue resolution (cf. Issue 14201) deprecates that change.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13958

Title: Section 7.2.3 Line 38-39 starting with "note that events" doesn't read

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 18 / line 37
Line 38-39 starting with "note that events" doesn't read

Resolution:

Clarify the text

Revised Text:

Page 19 / line 30
Change the whole paragraph from:
Note that Events CCM interface is never used in the connector's executor. The reason is that the connector' fragment and the component itself, are only interacting via synchronous calls as they are collocated. The actual interaction semantics between components is carried by connector' fragments themselves.
→
Note that the Events CCM interface is never used in connectors' executors. The reason is that, as the component and its connector's fragment are collocated, they only interact via synchronous calls (a potential asynchronous nature of the actual interaction between components would be provided by connector's fragments themselves).

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13959

Title: Section 7.2.4.2.1

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 21 / line 2
remove - from ConnectorPackage-Description Fix type
Page 21 / footer 3
in footer 3, ComponentUsageDescription and ComponentPackage Description (typo in first, space in second)

Resolution:

Correct the typos

Revised Text:

Page 21 / line 29
Change: ConnectorPackage-Description → ConnectorPackageDescription

Page 21 / in the footnote
Change: ComponentUsageDesription → ComponentUsageDescription
Change: ComponentPackage Description → ComponentPackageDescription

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13960

Title: Section 7.2.4.4 Line 3-4 doesn't read

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 25 / lines 3-4
doesn't read

Resolution:

Clarify the text

Revised Text:

Page 26 / lines 3
Change the whole paragraph:
Each fragment of a connector providing an implementation of an extended-port, have to be linked to its dual one, the mirror-port. In some cases, to be configured and linked together, fragment to configure have to query some data from the dual one. Taken into account that, the two fragments can be installed on different nodes, the process of configuration has to be remote.
→
All fragments of a given connector are in relation and have to be configured consistently. In some cases, this could require them to share configuration information that cannot be set statically. This dynamic initialization, if required, is connector implementation-specific and thus not specified. However it has to be completed by the end of the 'configuration_complete' phase of CCM deployment.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13961

Title: Section 7.2.4.5 this seems to lack a figure, line 7 ends with "extrat below" but nothing is below line 7

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 26
this seems to lack a figure, line 7 ends with "extrat below:" but nothing is below line 7

Resolution:

Clarify the text

Revised Text:

Page 27/ lines 6
Change the whole paragraph:
In addition to the ExtendedPortDef and ExtendedPortType, the concept of connector is introduced, represented in the meta-model extract below:
→
Finally, ConnectorDef is a new construct of the Component meta-model allowing modeling of connectors.
All those extensions are represented in figure 11.

Disposition: Resolved

Disposition: Resolved

OMG Issue No: 13963

Title: why not use an exception instead of a return value for the methods in Getter<>?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 33
why not use an exception instead of a return value for the methods in Getter<>?
On line 28, it should say "return as result" when decided to keep bool.
For the time_out, what about making this an argument of all methods, is setting this at this level at the correct level?

Resolution:

Proposed solution

1. keep the boolean as result and correct the text line 28
2. let the time_out as an attribute as purpose is to simplify as much as possible the API . Reducing the number of parameters goes in that direction. We noticed in our systems that time-out value was unlikely to change from one call to another, making it a recurring value is therefore sensible;

Revised Text:

Page 36 / line 20
Change the sentence:
They all receive as result a boolean that indicates
→
They all return a boolean as result indicating

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13964

Title: Section: 8.2.2.1.2 font

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 34 / line 11,12,13
The font of line 11,12,13 should show the name of the methods in bold
Page 34 / line 29
On_data should start with lower capital

Resolution:

Correct the typos

Revised Text:

Page 34 / line 11,12,13
Change the font of the name of the methods à bold

- 11: on_creation
- 12: on_update
- 13: on_deletion

Page 34 / line 29
Change: On_data → on_data

However, further issue resolution (cf. Issue 14214) deprecates that change.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13965

Title: Section 8.3.1 On line 19, after DDS_Base it should be a {

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 37 / line 19
after DDS_Base it should be a { instead [

Resolution:

Correct the typo

Revised Text:

Page 41 / line 2
Change: [→ {

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13966

Title: Section 8.4.2 On line 1, the annex numbers are lacking

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 39 / line 1
the annex numbers are lacking

Resolution:

Insert correctly the cross-references to the annexes.

Revised Text:

Page 42 / line 24 →
default values QoS policies, as specified in [DDS], are in Annex C: and Annex D: respectively.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13967

Title: Section 8.4.2 For Table 12, Boolean, font of BOOLEAN_TRUE is not ok

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 39 / line 12
For Table 12, Boolean, font of BOOLEAN_TRUE is not ok

Resolution:

Correct the typo

Revised Text:

Page 43 / in table 4
Change the character style of the last character of BOOLEAN_TRUE to the proper one (keyword)

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13968

Title: Section 8.2.2.2.5 On line 5 DURATION_INFINITE_NSEC should be used

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 42 / line 5
DURATION_INFINITE_NSEC should be used
Proposed solution:

Resolution:

Correct the typo

Revised Text:

Page 45 / line 40
Change: DURATION_INFINITE_SEC → DURATION_INFINITE_NSEC

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13970

Title: Section 9.2.1.1 line 39 should be indented

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 46 / line 39
Should be indented

Resolution:

Correct the typo

Revised Text:

Page 49 / line 39
Insert a TAB before raises

Page 58 / line 38
Insert a TAB before raises

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13971

Title: Section 9.2.2 Line 13 says DCPOS, shouldn't this be DCPS?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 47 / line 13
says DCPOS, shouldn't this be DCPS?

Resolution:

Correct the typo

Revised Text:

Page 50 / line 13
Change DCPOS → DCPS

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13972

Title: Section 9.2.2 Line 16 doesn't read, line 33 should also be clear

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 47 / line 16
doesn't read
Page 55;47 / line 33
should also be clear, "is likely" should be specified more strict

Resolution:

Clarify the text

Revised Text:

Page 51 / line 16
Change the sentence:
A DLRL connector is also essentially variable in its composition for it contains as many mirror ports as there are different object models to share the related topics. It...
→
As a DLRL connector aims at gathering as many mirror ports as there are different object models in the system sharing the related topics, its composition is essentially variable and application-dependent and a unique standard DLRL connector cannot be defined. A DLRL connector...

Page 51 / line 33
Remove likely and add at the end of the sentence:
in case there is a need to specify different QoS values for different topics.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13973

Title: Shouldnh't InternalError use DDSReturnCode_t

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 48
Shouldn't InternalError use DDS::ReturnCode_t
Proposed solution:
Page 56;48 / line 25
Replace unsigned long à DDS::ErrorCode_t

Resolution:

Apply change

Revised Text:

Page 53 / line 6
Change: `unsigned long` → `DDS::ErrorCode_t`

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 13974

Title: Annex D Font of line 20 not ok

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 56 / line 20
Font of line 20 not ok

Resolution:

Correct the typo

Revised Text:

Line 67 / line 13
Change the style of the line to the proper one(IDL Last)

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14017

Title: Section 8.2.2.1.4 and annex A missing parameter name

Source:

Thales, Virginie Watine, virginie.watine@thalesgroup.com

Summary:

Page 35 / line 33 and page 52 / line 47
Last parameter of ConnectorStatusListener:: on_unexpected_status (
    in DDS::Entity the_entity,
    in DDS::StatusKind);
should be given a parameter name

Resolution:

Correct the error

Revised Text:

Page 39 / line 18
Page 59 / line 19
Add status_kind as parameter name, new definition is

```
on_unexpected_status (
            in DDS::Entity the_entity,
            in DDS::StatusKind status_kind);
```

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14024

Title: keywords that can't be used in other IDL anymore

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The spec adds typename, primitive, port, mirrorport, porttype, and connector as keywords meaning that they can't be used anymore in other IDL. Below a list of files for TAO that have port, all from CORBA specs and for the keyword port. * orbsvcs/orbsvcs/SSLIOP.idl: * orbsvcs/orbsvcs/HTIOP.idl: * orbsvcs/orbsvcs/CSIIOP.idl: * tao/EndpointPolicy/IIOPEndpointValue.pidl: * tao/Strategies/sciop_endpoints.pidl: * tao/IIOP_Endpoints.pidl: * tao/IIOP.pidl: Because of this we need to use the _ prefix mechanism as specified in CORBA 7.2.3.1. To our idea the DDS4CCM spec should clearly specify all new keywords together and refer to the section in the corba spec. At the end also the CORBA spec has to be updated to list the new keywords

Resolution:

Gather all the new keywords in a table.

Revised Text:

Page 18 / line 21
Insert new section 7.3.6:

7.3.6 Summary of New IDL Keywords

The following table gathers all new keywords introduced by this specification.

Table 1: New IDL Keywords

alias	**connector**	**mirrorport**	**port**	**porttype**	**typename**

As all IDL keywords, they are now reserved and thus may not be used otherwise, unless escaped with a leading underscore.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14025

Title: table 2 on page 16 (adobe number) on line 1 should have the same fonts as table 1

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 8 / table 2
table 2 on page 16 (adobe number) on line 1 should have the same fonts as table 1

Resolution:

Correct the typo

Revised Text:

Page 16;8 / table 2
Interface, typename **|** struct **|** eventtype **|** primitive **|** fixed **|** sequence **|** interface **|** valuetype → in bold
interface_type → in italics

Page 8 / table 3
Porttype → port_type (2 times)

However, further issue resolution (cf. Issue 14201) deprecates that change.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14117

Title: ListenerControl

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 34
First wording : The DDS4CCM spec defines the interface ListenerControl. This is used for the various ports of 8.2.2.2. Some ports (like DDS_RawListen) do specify multiple listen ports, listener and status. There is now just one flag to enable/disable them both. maybe I am just interested in the status and not listener. We propose to move the attribute enabled to each listen interface and remove the ListenerControl interface from the spec. Then each listener can be enabled/disabled independently of each other
Second wording : we propose to change on page 36: uses ListenerControl control to uses ListenerControl listener_control and add to the ports that also have a provides PortStatusListener listener: uses PortStatusListener status_control;

Resolution:

The data listening extended ports are defined based on the fact that very often (not to say all times) an application component which wants to be notified of modifications of data (listener mode) needs to start by an initialization phase where he gets all the existing data (and not receive thousands of notifications to get the data that were existing before its creation). It is why they combine a Data Listener basic port with a Reader one. Before that init phase is completed, the component cannot treat well incoming notifications. This is why that DataListenerControl port is also provided to turn on/off the data 'tap'.
In return, Status Listeners are slightly different in nature: they are there to allow the component receive errors. The component has no choice with that and should be in position to get errors even during the init phase! It is the reason why those are always enabled (enabled at creation and no means / needs to enable them explicitly).

The recasting of DDS listen ports has added the necessity for two enabled modes (ONE_BY_ONE and MANY_BY_MANY). In addition, the StateListener is given a specific control interface allowing in addition to control how filtering in and out is reflected in the notifications.

Revised Text:

Page 37/ line 26
Replace the whole section 8.2.2.1.3 →

8.2.2.1.3 Data Listener Control

The following interface allows controlling the data listener attached to the port to which they are attached. There are two data listener controls:

- DataListenerControl which embed the basic controlling behavior for any kind of data listeners;
- StateListenerControl which is a specialization of the former which add extra feature for a StateListener.

Interface DataListenerControl

```
enum ListenerMode {
      NOT_ENABLED,
      ONE_BY_ONE,
      MANY_BY_MANY
      };
```

```
local interface DataListenerControl {
      attribute ListenerMode      mode;                 // default NOT_ENABLED
      attribute DataNumber_t      max_delivered_data;   // default 0 (no limit)
      };
```

The two attributes of a DataListenerControl allows controlling the associated data listener as follows:

- If the mode is NOT_ENABLED, the associated listener's operations are not triggered. This is the default setting as it allows the component to perform its initialization phase (likely using the associated Reader) before receiving any data notifications.
- If the mode is ONE_BY_ONE, the unitary operations (i.e. on_one_data or on_one_update) of the associated listener are triggered
- If the mode is MANY_BY_MANY, the grouped operations (i.e. on_many_data or on_many_updates) of the associated listener are triggered. These operations are called with as many relevant samples as available, possibly limited by the value of max_delivered_data. The default value for that attribute is UNLIMITED (0).

StateListenerControl

```
local interface StateListenerControl : DataListenerControl {
        attribute boolean                  is_filter_interpreted; // default FALSE
        };
```

This listener control, specific to control a StateListener, extends the former DataListenerControl with the attribute is_filter_interpreted.

- If TRUE, the associated listener should consider an instance entering in (resp. going out) the filter (if any) of the related Reader, as an instance creation (resp. deletion) and thus trigger the operation on_creation (resp. on_deletion).
- If FALSE, those events should be considered as normal instance updates and thus lead to triggering on_one_update or on_many_updates, depending on the mode.

Note: DDS is not currently reporting that an instance has been filtered out. This behavior has been thus added for provision. A compliant implementation of this specification is not required to support it as long as DDS does not report when instances are filtered out.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14168

Title: We propose to change the tags to <dds> and </dds>

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 39
On line 11 it says: The root tag of the configuration file must be <dds_ccm> and end with </dds_ccm>. We propose to change the tags to <dds> and </dds>. The XML QoS is coming from this spec but shouldn't be tied to it, it can be used without ccm at all.

Resolution:

Apply change

Revised Text:

Page 42 / line 2
Replace: dds_ccm → dds (2 times)

Page 66 / line 7
Page 69 / line 12
Replace: dds_ccm → dds

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14174

Title: One line 24 the spec uses BadParameter, but it doesn't describe when this exception has to be thrown

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 32
One line 24 the spec uses BadParameter, but it doesn't describe when this exception has to be thrown. On line 8 a void query is mentioned, but what is a void query? A string of size 0?

Resolution:

Clarify that the test of validity for query and query parameters is to be done by DDS.

Revised Text:

Page 35 / line 2
Replace the last sentence:
A void query means no query
→
An empty string query means no query.
This query and its related parameters are for DDS use and must comply with DDS rules (c.f. DDS specification for more details). Any attempt to set the attribute with values that are not accepted by DDS will result in a InternalError exception.

Remove the BadParameter exception in the IDL description.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14175

Title: if no key is specified the an_instance is supposed to be empty?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 32
On line 34 and 36 the spec mentions that the parameter an_instance has to be filled with the key value, but what if the topic doesn't have a key? I think the spec should mention this, if no key is specified the an_instance is supposed to be empty?

Resolution:

Clarify the test

Revised Text:

Page 35 / line 40
Add the following sentence at the end of the line:
In case of a keyless topic, the last value in the topic will be returned as DDS considers all values in such a topic as samples of one unique instance.

Page 36 / line 2
Add the following sentence at the end of the line:
In case of a keyless topic, all values will be returned as DDS considers all values in such a topic as samples of one unique instance.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14176

Title: line 17 and 21 the exception NonExistent: under which conditions has this exception to be thrown?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 32
On line 17 and 21 the exception NonExistent is mentioned but this chapter doesn't describe under which conditions this exception has to be thrown. What should the read_one and read_one_history do when there are no samples? Throw an exception? what is an_istance then with read_one?

Resolution:

As part of the DDS-DCPS ports reorganization, improve the description of the Reader operations has been improved

Revised Text:

Page 35 / line 33
Replace the description of the targeted operations (read_one_last and read_one_all) →

- read_one_last returns the last sample of a given instanceThe targeted instance is designated by the passed instance handle (instance_handle) if not DDS::HANDLE_NIL or by the key value in the passed data (datum) otherwise. If a valid handle is passed, it must be in accordance with the key value of the passed data otherwise an InternalError exception is raised with the returned DDS error code. More generally, any DDS error when reading the data will be reported by an InternalError exception.
  In case the instance does not exist (no data are registered for that instance in DDS), the exception NonExistent is raised.
  In case of a keyless topic, the last value in the topic will be returned as DDS considers all values in such a topic as samples of one unique instance.

- read_one_all returns all the samples of a given instance The targeted instance is designated by the passed instance handle (instance_handle) if not DDS::HANDLE_NIL or by the key value in the passed data (datum)

otherwise. If a valid handle is passed, it must be in accordance with the key value of the passed data otherwise an InternalError exception is raised with the returned DDS error code. More generally, any DDS error when reading the data will be reported by an InternalError exception.
In case the instance does not exist (no data are registered for that instance in DDS), the exception NonExistent is raised.
In case of a keyless topic, all values will be returned as DDS considers all values in such a topic as samples of one unique instance

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14177

Title: For timestamp, is this the DDS source_timestamp? If yes, rename the member to source_timestamp

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 31
For timestamp, is this the DDS source_timestamp? If yes, rename the member to source_timestamp

Resolution:

Change the attribute name

Revised Text:

Page 33 / line 29, 38,
Page 52 / line 34
Replace: `timestamp` → `source_timestamp`

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14178

Title: For line 44 and 20 I would recommend to have update/create/delete as bold font

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 30
For line 44 and 20 I would recommend to have update/create/delete as bold font, this refers to method names. Maybe split it in 2 sentences. For example: If is_lifecycle_checked is TRUE, then create checks that the instances are not already existing and update and delete that the instances are existing; AlreadyCreated and NonExistent exceptions may be raised. Change If is_lifecycle_checked is TRUE, then create checks that the instances are not already existing. Update and delete check that the instances are existing. AlreadyCreated and NonExistent exceptions may be raised.
Page 39 line3,
it says: The write or dispose orders are stopped. Write/dispose is DDS terminology, use: The update and delete calls are stopped ...
Page 39 line 3,
say what happens with the create call? Does it stop on the first error?

Resolution:

Clarify the text

Revised Text:

Page 30 / line 2
Replace:
the write orders are embedded between DDS begin_coherent_updates and a end_coherent_updates
→
the write DDS orders are placed between a DDS begin_coherent_updates and an end_coherent_updates

Page 30 / line 4
Change: write orders → write DDS orders

Page 30 / line 20
Change the font for create, update and delete to bold

Page 30 / lines 45,46
Change the font for create, update and delete to bold

Page 30 / line 47:
Replace: attempt to write or dispose → attempt to order DDS to write or dispose

Page 31 / line 2
Replace:
the write orders are embedded between DDS begin_coherent_updates and a end_coherent_updates
→
the write or dispose DDS orders are placed between a DDS begin_coherent_updates and an end_coherent_updates

Page 31 / line 4
Replace: write or dispose orders → write or dispose DDS orders

However, further issue resolution (cf. Issue 14214) deprecates some of those changes.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14179

Title: Line 14, change inees to fields

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 34 / line 34
Line 14, change inees to fields

Resolution:

Correct typo

Revised Text:

Page 37 / line 16
Change: inees → ones

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14201

Title: Destination module for implied template interfaces

Source:

Vanderbilt University, William Otte, wotte@dre.vanderbilt.edu

Summary:

Suppose we have this:

```
======== foo.idl:
module Foo {
    interface Foo_Intf < typename T >
        {    };
    };
 ========= bar.idl:
#include "foo.idl"
module Bar {
    porttype Bar_Port <typename T>
        {
         uses Foo_Intf<T> foo_port;
        };
    };
========= baz.idl:
#include "bar.idl"
module Baz {
    struct Baz_Struct
        {    };
    };
 ========= foobarbaz.idl:
    #include "baz.idl"
    module FooBarBaz {
        connector FooBarBaz_Connector     {
            mirrorport Bar_Port < Baz::Baz_Struct > foobarbaz_port;
            };
    };
 =========
```

In which module(s) are the implicit interfaces for Bar_Port and   Foo_Intf (with respect to their parameterization with Baz_Struct) assumed to be declared?

Resolution:

To give names to the instantiated interfaces, the initial version of the specification was proposing a naming convention which was far from covering all the cases as shown by this issue. In addition, this strategy was not in line with the one selected by IDL to get rid of anonymous types resulting from instantiation of existing templates (such as sequences). IDL now requires an explicit instantiation with a name allocated by the developer.

It has thus been decided to remove that naming convention and to rely on explicit instantiations.

Ina addition, a typical use case for parametrized extended ports and connectors is as follows: a parametrized interface is used in a similarly parametrized port type, which itself is used in a similarly parametrized connector. Eventually, when instantiated, it is mandatory that the concrete interface in support of the extended port attached to a component be exactly the same type as the one if support to the corresponding mirror port of the connector (otherwise connections will not be feasible), when in fact two explicit interface instantiations would result in two different types (even if identical). Therefore it is mandatory that the interface, the port type and the connector be in the same template scope, to allow a unique interface instantiation be shared by the instantiated port type and connector.

This lead to proposing template support for the only grouping structure of IDL, namely the module. Adding template modules is on the one hand less intrusive in the existing IDL grammar and on the other hand far more versatile than the initial proposed support. As it is needed to support almost all the possible extended ports and connectors and enough for that purpose, it was decided to limit the template support to template modules.

Consequently it was needed to recast almost entirely chapter 7, in order to present in one place the template support (while in the initial chapter organization, it was spread over the descriptions of each of the interaction constructs – port types, ports, connectors). The resulting outline is as follows:

7 Generic Interaction Support

7.1 Simple Generic Interaction Support

7.2 Generic Interaction Support with Templates

7.3 IDL3+ Grammar

7.4 Programming Model for Connectors

7.5 Connector Deployment and Configuration taking the opportunity of this recast, the whole grammar description has been rewritten in order to make it closer to the initial IDL grammar.

Revised Text:

Section 7.1, 7.2 and 7.3 are totally reorganized (cf. In the convenience document)

Section 7.4 is the initial section 7.2.3

Section 7.5 is the initial section 7.2.4

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14206

Title: Typo, erroneaous

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 48 / line 21
There is a typo in the batch 2 document, page 48, line 21: erroneaous

Resolution:

Correct typo

Revised Text:

Page 51 / line 57
Change: erroneaous → erroneous

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14207

Title: Typo, between

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 48 / line 45
In the spec there is a typo, between instead of betwen
Page 49, line 18,
or instead of od

Resolution:

Correct typo

Revised Text:

Page 53 / line 56
Replace between → between

Page 54 / line 36
Change: between → between

Page 54 / line 62
Replace od → or

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14209

Title: Dds4ccm and includes

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

I think it would be a good thing to standardize the IDL filenames where the dds ports are defined in to get interoperability in the user code.

Resolution:

Give the name “ccm_dds.idl” to the idl file.

Revised Text:

Page 30 / line 35
Add the following paragraph:
This IDL file is named "ccm_dds.idl".

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14210

Title: Reader port question

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

What is the behavior of the methods on the Reader interface when there is no data? Should the methods throw an exception or return a Boolean (but all read methods are void)

Resolution:

Clarify the text

Revised Text:

Page 34 / lines 35, 46
In the description of the operations Reader::read_one_last and read_one_all, add the following sentence:
In case the instance does not exist (no data are registered for that instance in DDS), the exception NonExistent is raised.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14212

Title: NonExisting::indexes with Reader/Getter/Writer

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

With the Reader/Getter/Writer we always handle one instance. When we can't find that instance, should NonExisting::indexes then just be a sequence that is empty, or with one index in it, 0?

Resolution:

Clarify that in this case, the sequence should contain the only index 0

Revised Text:

Page 33 / line 2
Add the following paragraph:
Note: In case of a single operation (create_one, update_one or delete_one) failing on the life cycle check, the sequence parameter of the exception (AlreadyExisting or NonExistent) will contain 0.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14213

Title: Sequence typedef leads to multiple sequences

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Several of the DDS ports have the following typedef: typedef sequence<T> T$Seq;   This leads to an unique sequence type for each DDS interface (MultiWriter, MultiUpdater, Reader, Getter, MultiListener). This leads to 5 times typecodes, footprint, but also the sequences are really from a different type in the programming language. A sequence returned from a Reader, just can't passed directly into a writer, it are different sequences.   I think we do want to have the same sequence to be used, the one related to T and this then has to be in the same namespace of T (the topic).
If we for example have
struct GPS {   long x;   long y; }
We want to get sequence<GPS> GPSSeq in the global namespace.   Not ::CCM_DDS::GPS_reader::GPSSeq, ::CCM_DDS::GPS_getter::GPSSeq, etc.
I think we should remove the typedef of the interfaces and add it as a template argument, so that for example we get a reader below.

```
interface Reader <typename T, typename TSeq> {
        void read_all (out TSeq instances, out ReadInfoSeq infos)
                raises (InternalError);
        void read_all_history (out TSeq instances, out ReadInfoSeq infos)
                raises (InternalError);
        void read_one (inout T an_instance, out ReadInfo info)
                raises (NonExistent, InternalError);
        void read_one_history (in T an_instance,
                out TSeq instances, out ReadInfoSeq infos)
                raises (NonExistent, InternalError);
        attribute QueryFilter filter
                setraises (BadParameter);
        }
```

This would mean that the user of the interface/port has to instantiate it with the basic type and the sequence type, but that seems the only way to get only one sequence definition used between all the ports, corba, and dds itself. The sequence just also has to be in the same namespace as the original type T, we just can't do that with a typedef in an interface that uses the sequence.

Resolution:

The whole template support has been revisited. According to that new support all parametrized constructs are now grouped in a module. The CCM_DDS::Typed module is parameterized with T and Tseq – sequence <T>).

Making the second parameter of type sequence<T> allows to check at compile time that the type of the second parameter is actually sequence of the first one.

Revised Text:

Page 29 / line 6
Change the paragraph →
DDS-DCPS ports and connectors will be grouped in a module, itself parameterized by the data type and a sequence type of that data type.

- Grouping the definitions for port types and connectors in the same module allows that they share the same concrete interface when eventually instantiated.

- Passing that second parameter may seem redundant but it is the only way to allow sharing the sequence definition with the rest of the application[10].

To avoid useless duplications when instantiated, this template module will only contain the constructs that depend on the data-type. It will be included in a more general module that will also contain all the constructs that do not depend on the data-type.

[10] Otherwise, the sequence created by this definition would be a type different (even if actually identical) from the one used by the application (created by the application or by DDS), which would lead to continual copies between one and the other.

Page 30 / line 23
Add the following text:

All those constructs are included in the Typed template sub-module of the CCM_DDS module, as follows:

```
module CCM_DDS {
      // Non-typed definitions
      ...
      module Typed <typename T, sequence<T> TSeq> {
            // Typed definitions
            …
            };
      };
```

In the following sections are thus listed extracts from the template module CCM_DDS::Typed<typename T, sequence<T> Tseq>.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14214

Title: Do we need the Multi* interfaces/ports?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

I have been thinking of how I could use the Multi* interfaces/ports. I can imagine that a component developer first starts to use the basic ports and when he is more concerned with efficiency/performance he wants to use the Multi ports where he can send batches of data to DDS in one call. This looks nice, but when he wants todo that, he has to use different ports, which leads to a different component design. It is just not an easy change to make a new implementation of the component, it will become something completely different by use the other ports. No easy swap to a more efficient component implementation.
I think we should remove all Multi ports and merge their functionality into the regular ports. That way a component developer can use the more efficient sequences based interfaces without large changes in models and deployment plants.

Resolution:

Merge the multi-operations with the simple ones in a consolidated interface:

Writer + MultiWriter → Writer
Updater + MultiUpdater → Updater
RawListener + MultiListener → Listener
StateLisener + MultiLisener → StateListener

Take the opportunity of this recast of the IDL description for the DDS ports, to better harmonize the names of the operations and to rationalize the roles of each interface.

Revised Text:

Page 31 to 39
Replace the whole section 8.2.2.1 DDS-DCPS Basic Port Interfaces. (Cf. Convenience document for the new text)

Page 39 / line 31
Replace the list of port types →

```
porttype DDS_Write {
      uses Writer                   data;
      uses DDS::DataWriter          dds_entity;
      };
porttype DDS_Update {
      uses Updater                  data;
      uses DDS::DataWriter          dds_entity;
      };
porttype DDS_Read {
      uses Reader                   data;
      uses DDS::DataReader          dds_entity;
      provides PortStatusListener   status;
      };
porttype DDS_Get {
      uses Reader                   data;
      uses Getter                   fresh_data;
      uses DDS::DataReader          dds_entity;
      provides PortStatusListener   status;
      };
porttype DDS Listen {
      uses Reader                   data;
      uses DataListenerControl      data_control;
      provides Listener             data_listener;
      uses DDS::DataReader          dds_entity;
      provides PortStatusListener   status;
      };
porttype DDS_StateListen {
      uses Reader                   data;
      uses StateListenerControl     data_control;
      provides StateListener        data_listener;
      uses DDS::DataReader          dds_entity;
      provides PortStatusListener   status;
      };
```

Page 40 / line 16
Change the whole paragraph →
All proposed DDS ports combine at least a basic port to access data with a basic port to access underlying DDS entity. DDS_Get, DDS_Listen and DDS_StateListen split the data access functionality in two ports; the first one (Reader) is there to set the read criterion and provide operations for the initialization phase, while the second one (Getter, Listener or StateListener) is rather intended to be used in the application processing loop. All the ports intended for the subscribing side comprise also a port to be notified of the relevant statuses.

Page 41 / line 26
Add the following bullet

- push_state_observer are being notified with the state changes with different operations depending on the instance status.

Page 41 / line 29
Replace the connector description →

```
connector DDS_State : DDS_TopicBase {
      mirrorport DDS_Update         observable;
      mirrorport DDS_Read           passive_observer;
      mirrorport DDS_Get            pull_observer;
      mirrorport DDS_Listen         push_observer;
      mirrorport DDS_StateListen    push_state_observer;
      };
```

Page 41 / line 3
Replace the connector description →

```
connector DDS_Event : DDS_TopicBase {
      mirrorport DDS_Write          supplier;
      mirrorport DDS_Get            pull_consumer;
      mirrorport DDS_Listen         push_consumer;
      }
```

Page 50
Replace the whole annex A with the one in the convenience document

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14567

Title: Updater and instance handle

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Currently DDS4CCM specifies the Updater port below. We are looking at using register_instance/write/unregister_instance for this? If this port is intended to deliver an abstraction of these methods, why don't we pass the instancehandle back to the CCM Level?

Resolution:

Add the optional instance_handle parameter to each operation dealing with a specific instance.

Revised Text:

Page 31 / line 10
Add the following section:

InstanceHandleManager

```
abstract interface InstanceHandleManager {
      DDS::InstanceHandle_t register_instance (in T datum)
            raises (InternalError);
      void unregister_instance (in datum, in DDS::InstanceHandle_t 
instance_handle)
            raises (InternalError);
      };
```

This interface gathers the two operations that allows manipulating DDS instance handles and will serve as a basis for the Writer or the Updater interfaces.

- register_instance asks DDS to register an instance, which results in allocating it a local instance handle. The targeted instance is indicated by the key value in the passed data (datum).

- unregister_instance asks DDS to unregister the instance, indicated by the passed instance_handle and thus to release the instance handle.

Both operations are very similar to the DDS ones and are just passed to the DDS DataReader in support for the relater DDS port. Cf. the DDS documentation for more details. Any DDS error will be reported through an InternalError exception.

Page 31 / line 25

Replace the beginning of Writer definition →

Interface Writer

```
local interface Writer : InstanceHandleManager {
      void write_one (in T datum,
                  in DDS::InstanceHandle_t instance_handle)
            raises (InternalError);
      void write_many (in TSeq data)
            raises (InternalError);
      attribute boolean is_coherent_write; // FALSE by default

      };
```

Behavior of a Writer is as follows:

- write_one allows publishing one instance value. The targeted instance is designated by the passed instance handle (instance_handle) if not DDS::HANDLE_NIL or by the key value in the passed data (datum) otherwise. If a valid handle is passed, it must be in accordance with the key value of the passed data otherwise an InternalError exception is raised with the returned DDS error code. More generally, any DDS error when publishing the data will be reported by an InternalError exception.

Page 31 / line 43
Replace the beginning of Updater definition → :

Interface Updater

```
local interface Updater : InstanceHandleManager {
      void create_one (in T datum)
            raises (AlreadyCreated,
                  InternalError);
      void update_one (in T datum,
                  in DDS::InstanceHandle_t instance_handle)
            raises (NonExistent,
                  InternalError);
      void delete_one (in T datum,
                  in DDS::InstanceHandle_t instance_handle)
      raises (NonExistent,
                  InternalError);
      void create_many (in TSeq data)
            raises (AlreadyCreated,
                  InternalError);
      void update_many (in TSeq data)
```

```
        raises (NonExistent,
              InternalError);
void delete_many (in TSeq data)
        raises (NonExistent,
              InternalError);
readonly attribute boolean is_global_scope; //FALSE by default
attribute boolean is_coherent_write; // FALSE by default
};
```

Behavior of an Updater is as follows:

- create_one (resp. update_one, delete_one) allows creating (resp. updating, deleting) one instance. For create_one this instance is designated by the key value in datum. For the two others, it is designated by the passed instance handle (instance_handle) if not DDS::HANDLE_NIL or by the key value in the passed instance data (datum) otherwise. If a valid handle is passed, it must be in accordance with the key value of the passed instance data otherwise an InternalError exception is raised with the returned DDS error code. More generally, any DDS error when publishing the data will be reported by an InternalError exception.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14574

Title: topic_name attribute

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

we propose to change the topic_name attribute to a regular attribute, not readonly.

Resolution:

This attribute as well as the others on DDS-DCPS connectors are not changeable dynamically as they are used to create and configure the DDS entities in support to this DDS pattern. However as their configuration by an external tool is highly desirable, they have been turned into regular attributes which can raise a NonChangeable exception.

Revised Text:

Page 40 / line 37
Change section 8.3.1 text →

DDS_Base connector uses a ConnectorStatusListener port for reporting configuration errors and contains attributes to store the Domain identifier and the QoS profile (c.f. section 8.4.2 for more details on QoS profile). The QoS profile could be given either as a file URL or as the XML string itself.

Any attempt to change those attributes once the configuration is complete will raise a NonChangeable exception.

All DDS connectors should inherit from that base.

```
connector DDS_Base {
      uses ConnectorStatusListener  error_listener;
      attribute DDS:DomainId_t            domain_id
            setraises (NonChangeable);
      attribute string        qos_profile // File URL or XML string
            setraises (NonChangeable);
      };
```

DDS_TopicBase extends the DDS_Base with the name of one topic and its key description. DDS_TopicBase should be the base for all mono-topic connectors

```
connector DDS_TopicBase : DDS_Base {
      attribute string                    topic_name
            setraises (NonChangeable);
      attribute DDS::StringSeq            key_fields
            setraises (NonChangeable);
      };
```

As the attributes of DDS_Base, the attributes of DDS_TopicBase are also non changeable once configured. Any attempt to change them once the configuration is complete will raise a NonChangeable exception.

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14590

Title: DDS port interfaces should be local

Source:

Thales, Virginie Watine, virginie.watine@thalesgroup.com

Summary:

The DDS4CCM specification states that a component with a DDS port and the fragment of connector supporting this DDS port should be collocated but the related interfaces are not defined as local.

Resolution:

Define all interfaces used in the DDS ports as local

Revised Text:

Page 31 / lines 26, 44
Page 35 / line 4
Page 36 / lines 5, 27
page 37 / line 35
Page 38 / lines 7, 32
Page 39 / line 2
Page 49 / line 8
Page 52 / lines 23, 28, 34, 45
Page 53 / lines 47, 62
Page 54 / line 44
Page 55 / lines 14, 38, 52
Page 57 / line 7
Replace: `interface` → `local interface`

Disposition: **Resolved**

Disposition: Resolved

OMG Issue No: 14611

Title: Module name

Source:

Thales, Virginie Watine, virginie.watine@thalesgroup.com

Summary:

In the specification, the IDL module name is sometimes "DDS_CCM" sometimes "CCM_DDS". It should be "CCM_DDS" everywhere.

Resolution:

Nme always the module CCM_DDS

Revised Text:

Page 50 / lines 5, 22
Page 58 / line 5
Replace: `DDS_CCM` → `CCM_DDS`

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

Disposition: **Deferred**

Disposition: Deferred

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.

Discussion:

No time to tackle it completely

Disposition: **Deferred**

Disposition: Closed, no change

OMG Issue No: 13962

Title: Section 8.2.2.1.1 what about making is_lifecycle_checked a regular attribute, so not read only?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 30
what about making is_lifecycle_checked a regular attribute, so not read only?

Discussion:

As stated in the spec, checking the lifecycle is quite an heavy process as it may imply to create a reader for that purpose. In addition, whether this behavior has to be enforced is unlikely to change over the component's life. For those reasons, the behavior has to be set at init time and in not changeable after that.

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14080

Title: propose to read the AMI chapter that was part of the alpha spec

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

We propose to add the AMI chapter that was part of the alpha spec. It seems useful for ccm to also standardize the ami ports. the chapter will need some more work and fine tuning

Discussion:

As its title says, this specification is aiming at standardizing ports for DDS and proposes a general framework to add ports (GIS – Generic Interaction Support), then uses it for DDS. Standardizing AMI ports is out of RFP scope.

This example was provided in the submission only as a rationale for the GIS. It was never intended to be normative.

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14110

Title: SampleInfo

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The CCM4DDS spec defines SampleInfo in its own IDL. It describes that this is a simplified version of DDS::SampleInfo. This new struct has the following disadvantages to our idea: - the struct that gets received from DDS itself has to be converted to the CCM4DDS struct taking performance - not all DDS information is available anymore for the component We propose to use DDS::SampleInfo throughout the spec instead of introducing a simplified shadow struct

Discussion:

The spec intended to ease use of DDS. Many DDS users feedback show that they can be easily lost with some SampleInfo information. SampleInfo is the pure view on the internal state of the data. ReadInfo attempts to translate it a more user-centric view. Note that translation SampleInfos in ReadInfos is not only copying a few fields from the firsts to the seconds: it requires also combining them.

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14208

Title: Reader interface

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

I noticed that all methods on the Reader<typename T> interface have only out arguments. Some return all samples for all instances, with just out, the caller can't preallocate a buffer that can be used, the memory has to be allocated each time again and released when not used. This doesn't really use the power of DDS to my idea, any ideas on this?

Discussion:

Purpose is to get IDL simple interfaces.

In case performance is really a huge concern, the DDS basic port can be used

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14217

Title: QoS profile attribute name?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The DDS4CCM spec says the following about specifying qos profiles through XML *** A QoS Profile shall be attached as a configuration attribute to a DDS connector. This profile should contain all values for initializing DDS Entities that are required by the connector. *** Shouldn't the attribute name also be standardized?

Discussion:

The name is given in the DDS_Base connector description (qos_profile)

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14558

Title: Extended ports

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

On page 5, line 17, the spec says the following of extended ports:
Extended ports allow grouping a set of single CCM ports (facet/provides and receptacle/uses), to define a particular semantic. The extended ports are declared in IDL3+ (extended IDL3 with new keywords). A new keyword is introduced to define extended ports (porttype) and to declare an extended port at component level (port).

Shouldn't this be:
Extended ports allow grouping a set of single CCM ports (facet/provides and receptacle/uses) and extended ports, to define a particular semantic. The extended ports are declared in IDL3+ (extended IDL3 with new keywords). A new keyword is introduced to define extended ports (porttype) and to declare an extended port at component level (port).

So, add that an extended port can contain extended ports again.

Discussion:

Currently it is not planned that an extended port be recursive. The reason is that while we fail to see any use case where that feature would really add something (apart being intellectually satisfactory), it brings an extra-complexity, not only at the CIF level (name of the resulting ports) but also at D&C tooling level.

Therefore it is decided not to change that feature.

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14578

Title: on_subscription_matched missing

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

on_subscription_matched is missing from the connectorstatuslistener, we propose to add this to one of the listeners

Discussion:

DDS states this feature as optional

In case it is really needed the provided port DDS entity allows accessing it

Disposition: **Closed, no change**

Disposition: Closed, no change

OMG Issue No: 14581

Title: readonly attribute DDS::StringSeq key_fields;?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Why does 8.3.1 add the following attribute to the DDS_TopicBase? What is the functional meaning of this, shouldn't we just remove this attribute?

readonly attribute DDS::StringSeq   key_fields;

Discussion:

There is currently no standard DDS way to indicate the key fields. Placing this information at the connector level is useful for system documentation.

Disposition: **Closed, no change**

Disposition: Duplicate/merged

OMG Issue No: 13885

Title: Shouldn't table 10 take parameterized connectors into account?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 16 / lines 8-9
Shouldn't table 10 take parameterized connectors into account?

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 13889

Title: change nb to number in the comment on line 29/32/35

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 30 / lines 29/32/35,
change nb to number in the comments

Disposition: **See issue 13888 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 13895

Title: Line 13/16/21/24, shouldn't the DDS_ be removed from the kind?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Line 13/16/21/24, shouldn't the DDS_ be removed from the kind?

Disposition: **See issue 13894 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 13897

Title: Line 15,19, use CORBAULongSeq instead of ULongSeq Line 23, use an enum for the error_code?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 48 / line 15,19
use CORBA::ULongSeq instead of ULongSeq Line

Page 48 / line 23
use an enum for the error_code?

Disposition: **See issues 13949 + 13973 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 13949

Title: The usage of ULongSeq should be CORBAULongSeq

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 48 / lines 15,19
The usage of ULongSeq should be CORBA::ULongSeq to show that these types are coming from the CORBA namespace

Disposition: **See issue 13888 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 13952

Title: this struct uses a StringSeq but the namespace is lacking

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 50 / line 12
struct QueryFilter uses a StringSeq but the namespace is lacking, should it be CORBA::StringSeq?

Disposition: **See issue 13890 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 13969

Title: Section 9 Line 4 should start with a capital

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 45 / line 4
Should start with a capital

Disposition: **See issue 13896 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14020

Title: The spec uses $ as string replacement for IDL3+

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 6 / line 40
The spec uses $ as string replacement for IDL3+. We find this not real and doubt whether a new string substitution should be added to the world. We would propose to use a different form, based on 'variable substitution' as found in unix shells, perl, ruby Two other options where we prefer the first one - use ${var} or #{var} or %var%; this is much clearer and matches already supported substitution options The new way would be: interface EventsPusher <typename T> { typedef sequence<T> #{T}Seq; void push (in #{T}Seq events); void push_#{T} (in T event); }; The other option is to use the C preprocessor way X##Y -> XY It then becomes interface EventsPusher <typename T> { typedef sequence<T> T##Seq; void push (in T##Seq events); void push_##T (in T event); };

Disposition: **See issue 13884 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14028

Title: The spec doesn't say anything about forward declared IDL3+ interfaces

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The spec doesn't say anything about forward declared IDL3+ interfaces. If it is not allowed, it should be mentioned. If it is allowed, the spec should mention it and describe it like section 7.8.4 of the regular CORBA spec for regular interfaces

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14037

Title: DDS4LwCCM - template interface IDL grammar

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

On page 11, line10, it appears that the (optional) inheritance spec comes after the interface body. This is probably not intended.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14038

Title: DDS4LwCCM - template interface body

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 11 / line 10
On page 11, line 10, the body of a template interface is given as identical to that of a non-template interface. However, if the concatenation symbol $ can be used in a template interface member, how can this be? The $ symbol can't be part of a legal identifier, so it will have to be a token in its own right. This changes the grammar at some level, whether it be in the definition of an individual export, such as an operation, or in the definition of an identifier itself.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14039

Title: DDS4LwCCM - forward declared template interface?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Nothing in the new IDL3+ production rules suggests that a template interface can be forward declared. This restriction makes sense if a template interface may be used only in an extended port definition (submitted as a separate issue), but it would be helpful to state the restriction explicitly in the document.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14040

Title: DDS4LwCCM - standalone template interfaces?

Source:

Vanderbilt University, Jeff Parsons, j.parsons@vanderbilt.edu

Summary:

Is it illegal to use a template interface to define an 'interface type' which may then be specialized for simple CORBA object requests? My guess would be that a template interface is intended for use only in an extended port, but it would be helpful to state it explicitly in the document.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14046

Title: DDS4LwCCM - concatentation symbol usage

Source:

Vanderbilt University, Jeff Parsons, j.parsons@vanderbilt.edu

Summary:

On page 7, lines 24-26, there are examples of $ usage. These all apply to the parameter value following the type classifier 'typename'. Will it also be legal to use it with parameter values following other type classifiers?

Disposition: **See issue 13884 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14057

Title: the spec should be more clear for which constructs the $ can be used

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

the spec should be more clear for which constructs the $ can be used. The rules for the regular IDL should be extended. Is this possible for attributes, exceptions, etc?

Disposition: **See issue 13884 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14119

Title: What are the semantics of the 'primitive' keyword? Letting it match any primitive type seems too vague

Source:

Vanderbilt University, Jeff Parsons, j.parsons@vanderbilt.edu

Summary:

Page 11
What are the semantics of the 'primitive' keyword? Letting it match any primitive type seems too vague

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14120

Title: The non-terminal 'extended_port_decl' appears as both a component export and a connector export.

Source:

Vanderbilt University, Jeff Parsons, j.parsons@vanderbilt.edu

Summary:

Page 12
The non-terminal 'extended_port_decl' appears as both a component export and a connector export. In particular, the 'generic_template_spec' non-terminal is problematic here. Since a component cannot be parameterized, as a component export, it must be the scoped name of a previously defined IDL type. As a connector export, however, it would appear to be legal IDL in this form, but also as a simple identifier matching one of the connector's template parameters. Are both constructs legal as connector exports? If so, they can't both be covered by 'generic_template_spec'.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14121

Title: On line 24, the term '<connector_export>' should be '<connector_export>+'.

Source:

Vanderbilt University, Jeff Parsons, j.parsons@vanderbilt.edu

Summary:

Page 17
On line 24, the term '<connector_export>' should be '<connector_export>+'.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14122

Title: Lines 32-36 imply that value declarations and type declarations may be parameterized

Source:

Vanderbilt University, Jeff Parsons, j.parsons@vanderbilt.edu

Summary:

Page 12
Lines 32-36 imply that value declarations and type declarations may be parameterized the same as interface declarations. By extension, lines 14 & 15 imply that value declarations and type declaration may also be used to implement 'provides' and 'uses' ports.

Disposition: **See issue 14201 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14118

Title: Section 8.2.2.1.3 ListenerControl attribute

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 34
First wording: The ListenerControl defines the attribute enabled. When checking the DDS spec I see that DDS only has the feature to enable an entity, not disable it. Shouldn't this attribute not be defined as: void enable (); as in the dds spec 7.1.2.1.1.7 enable Also DDS says an entity is enabled by default, the DDS4CCM says false in 8.2.2.1.3 What is the background of this, shouldn't DDS4CCM say it is enabled by default?
Second wording: We propose to change line 33 on page 34 from "The following interface allows to enable the listeners that are attached to the port." To "The following interface allows to enable and disable the listeners that are attached to the port."

Disposition: **See issue 14117 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14211

Title: non-keyed datatypes

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

When I look at the reader I get the impression this interface is designed with keyed data types in mind. How would for example read_one work when there is no key defined? What is the value that then should be passed in as instance?

Disposition: **See issue 14176 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14180

Title: MultiWriter

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 29
For MultiWriter we have: unsigned long write(in T$Seq instances) // returns nb of written raises (InternalError); But, when write fails we get an exception. So, when it success all data has been written and then the return value is not needed. I would recommend to make this method void. In case it succeeds all instances are written, if it fails, the exception will tell you the number failed, so it the number written is one less then the number failed

Disposition: **See issue 14182 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14181

Title: or MultiUpdater the return value of unsigned long doesn't make sense

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Page 30
Also for MultiUpdater the return value of unsigned long doesn't make sense, if the methods work, the number of elements written is all, if it fails, the exception will tell you the number failed

Disposition: **See issue 14182 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14182

Title: Return type of the MultiWriter/MultiUpdater?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

Why do the MultWriter/MultiUpdater methods have a return value of unsigned long and not void? When the method succeeds all elements are written, if it fails, the return value itself is worthless because we get an exception. When it has succeeded the number of instances written is the same as the number of instances passed, or not?e ??

Disposition: **See issue 14214 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14202

Title: Reader::read_all_history

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

The Reader::read_all_history is documented as: read_all_history returns all samples of all instances   This could return a lot of data, the reader has no idea how much he will get. Does this make sense? There is no way the Reader can limit the history to a certain size as we could do with DDS directly.

Disposition: **See issue 14214 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14534

Title: MultiListener

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

On page 34 (word) on line 30 it gives the paragraph below. There is an inconsistency there, it says "a sequence" and the next sentence it says "each sequence". Looking at the IDL, it just is one sequence, so it should probably be "Depending of grouping mode, the sequence will contain"

Disposition: **See issue 14214 for disposition**

Disposition: Duplicate/merged

OMG Issue No: 14562

Title: (Multi)Updater method names?

Source:

RemedyIT, Johnny Willemsen, jwillemsen@remedy.nl

Summary:

We noticed that the (Multi)Updater uses delete as method name, this is a reserved keyword in C++ which leads to _cxx_delete in the C++ mapping. Maybe it is an idea to use different method names, like register/write/dispose as in DDS? It is now a small change, but prevents users to use _cxx_delete when using C++.

Disposition: **See issue 14214 for disposition**