



WEBINAR

OpenDDS Town Hall

Panelists: Jen Wiese, Adam Mitz, Tim Simpson,
Justin Wilson, Fred Hornsey, and David James



March 31, 2022



The OpenDDS Foundation™ and Object Computing, Inc. (OCI)
12140 Woodcrest Exec. Dr., Ste. 300
Saint Louis, MO 63141 USA

© 2022 All Rights Reserved

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior, written permission of the OpenDDS Foundation and OCI.

Limits of Liability and Disclaimer of Warranty

While every precaution has been taken in preparing this material, including research, development and testing, the OpenDDS Foundation and OCI assume no responsibility for errors or omissions. No liability is assumed by either the OpenDDS Foundation or OCI for any damages resulting from the use of this information.



- Welcome and Introductions
- Updates and Upcoming Events
- Foundation Update
- Case Study Presentation
- OpenDDS Project Status Update
- Community Q&A



OpenDDS Town Hall Meeting

Gather to review the most up-to-date news and resources available, celebrate accomplishments, ask questions, and participate more fully in the OpenDDS ecosystem.



Online Training Offerings:

- May 2 - May 3
 - Introduction to OpenDDS Programming (C++, Java)
- May 4 - May 5
 - Building OpenDDS Applications with DDS Security (C++, Java)
- May 9 - May 10
 - OpenDDS Essentials I: (C++, Java)
 - QoS, Keys and Instances, & Built-In Topics
- May 11 - May 12
 - OpenDDS Essentials II: (C++, Java)
 - Configuration, Listeners, Conditions, & Content-Subscription



OpenDDS
FOUNDATION™

The OpenDDS Foundation is a not-for-profit organization that exists to support and collectively lead the open source OpenDDS project

The OpenDDS Foundation serves to:

- Ensure technical innovation and advancement of the OpenDDS project as a free and open public-use software development toolkit for a growing global community
- Evangelize and promote OpenDDS technology as a leading implementation of the Object Management Group's Data Distribution Service (DDS)
- Build and support an ecosystem of complementary documentation, functionality, and services



Stewardship

The OpenDDS Foundation is supported by a Technology Advisory Board (TAB). The TAB's missions are to gain required insights, so the OpenDDS Project can continue to evolve, and to ensure it continues to reflect and serve its diverse and growing user community.

Current TAB members are OpenDDS users and supporters from Washington University in St. Louis, Leonardo DRS, OCI, Plotlogic, United States Navy, and Remedy IT

The TAB meets quarterly to discuss and make recommendations regarding the Projects' roadmap and technical direction, as well as ongoing and future community engagement activities.





COMMUNITY

ENGAGEMENT

- Technical Training
- Up-To-Date Documentation
- Free Technical Webinars
- On-Demand Content
- Regular Publications of News Posts and Articles
- Social Media and Participation in Community Events
- Technical Support Options
- Industry Solutions and Case Study References
- Access source code, submit issues, contribute to the project, and ask questions here





Sponsorship

<https://opendds.org/foundation/sponsorship.html>

As a not-for-profit organization, the OpenDDS Foundation relies on the financial support of Contributing Members to support and grow the project.

- Corporate Sponsorship

The generous support of our Corporate Sponsors allows the OpenDDS Foundation to further the project's technical innovation and advancement.

[Learn more about the Corporate Sponsorship Program](#)

- Community Member Sponsorship

Individuals may also support the OpenDDS Foundation through our Community Member Sponsorship program.

[Learn more about Community Member Sponsorship](#)



David (DJ) James - Technical Director for Software, DRS Land Systems

Integrating OpenDDS Middleware Into Next-Generation Surveillance and Battlefield Reconnaissance Equipment

OpenDDS middleware enables leading provider of defense technologies to accelerate development processes and deliver powerful, cost-effective solutions to clients worldwide.



<https://objectcomputing.com/case-studies/drs-land-systems>

Summary of Releases in 2021



- 4 minor releases and 2 point releases in 2021
 - OpenDDS 3.16 Jan 27 XTypes
 - OpenDDS 3.17 May 19 JSON, RTPS logging
 - OpenDDS 3.18 Aug 24 IDL4 int types, RTPS performance
 - OpenDDS 3.16.1 Aug 27
 - OpenDDS 3.18.1 Sep 24
 - OpenDDS 3.19 Dec 10 XTypes Dynamic
- Latest is OpenDDS 3.20 March 25, 2022
- Plan is to continue a quarterly release cadence

<https://github.com/objectcomputing/OpenDDS/releases>



- Android
 - Support for Android API Level 30+ (Android 11)
 - This is the first time OpenDDS has used the Android Java API
 - Expanded ACE/TAO to support building with the Android NDK directly
- OpenDDS on the Internet
 - Made many stability and performance improvements to the RtpsRelay

Learn more about powerful IoT connectivity driven solutions with OpenDDS technology [here](https://objectcomputing.com/products/opendds/solutions/iot-connectivity)
<https://objectcomputing.com/products/opendds/solutions/iot-connectivity>



- C++
 - Support for IDL4 explicit integer types
 - Support for using IDL and C++ keywords in IDL
 - Small improvements to C++11 mapping and anonymous type support
- Java
 - Also got IDL4 explicit integer types
 - Memory improvements in generated JNI code for large IDL data structures



- node-opendds (Javascript)
 - Updated to support recent OpenDDS releases
 - Migrated CI to GitHub Actions
- PyOpenDDS (Python)
 - Updated to support recent OpenDDS and Python releases
 - Added support for Windows and macOS
 - Improvements to build system and testing
 - Started work on supporting publishing, basic QoS, and more types
- OpenDDSharp (C#/.NET – independent project)
<https://github.com/jmmorato/openddsharp>
 - Updated to work with recent OpenDDS releases
 - Added support for Visual Studio 2022 and macOS



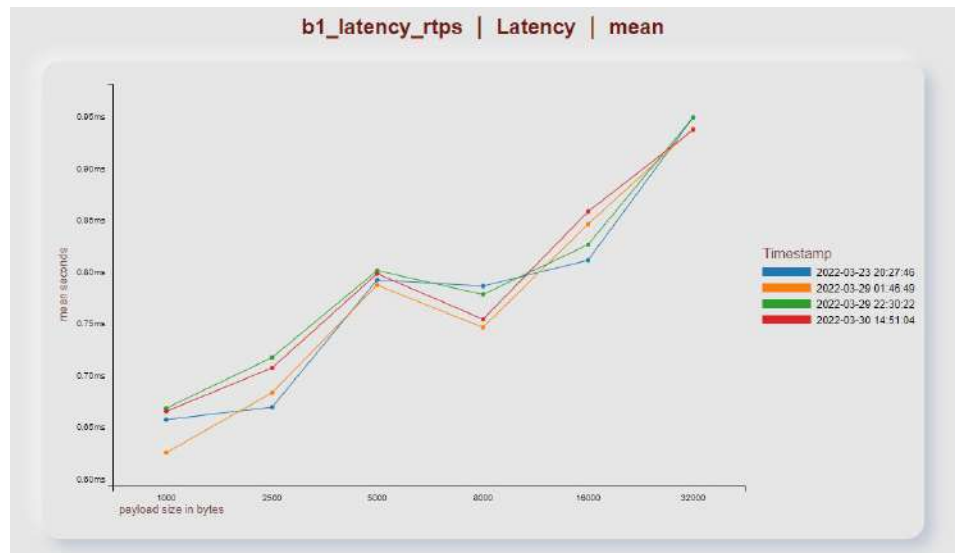
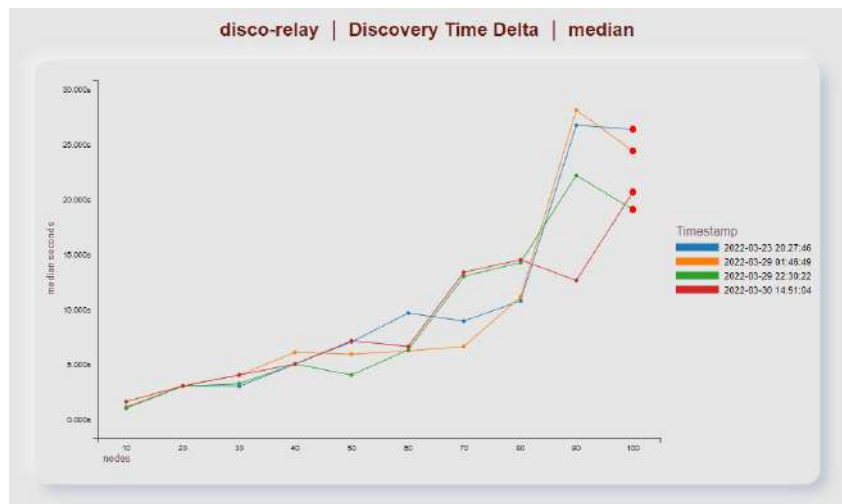
Previously (2020):

- Some CI Coverage:
 - Jenkins PR Builder Test Infrastructure
 - Some Travis CI / Azure Pipelines Coverage
- Post-Merge:
 - Jenkins Scoreboard for Full Testing & Address Sanitizer Checks
 - Performance & Stability Testing Dashboard (Bench)



Improvements (2021 - present):

- **Continued Improvements to Performance Dashboard**



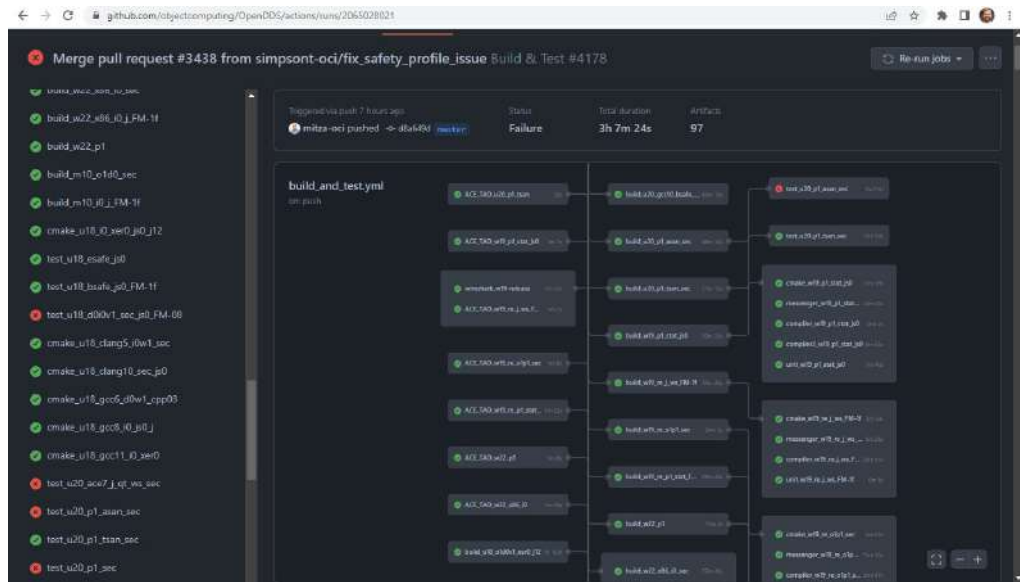
Scalability and Performance Testing with the OpenDDS Bench Framework: [On-Demand Webinar and Resources](https://objectcomputing.com/products/opendds/resources/bench-framework)

<https://objectcomputing.com/products/opendds/resources/bench-framework>



Improvements (2021 - present):

- Continued Improvements to Performance Dashboard
- **Move CI to GitHub Actions**
 - **30+ build configurations**
 - **22+ of which include tests**
 - **Address Sanitizer (full suite)**
 - **Thread Sanitizer (partial)**



Improved Usage of Development Tools (cont)



Improvements (2021 - present):

- Continued Improvements to Performance Dashboard
- Move CI To GitHub Actions
 - 30+ build configurations
 - 22+ of which include tests
 - Address Sanitizer (full test suite)
 - Thread Sanitizer (partial)
- **Unit Test Coverage Reports (LCOV)**

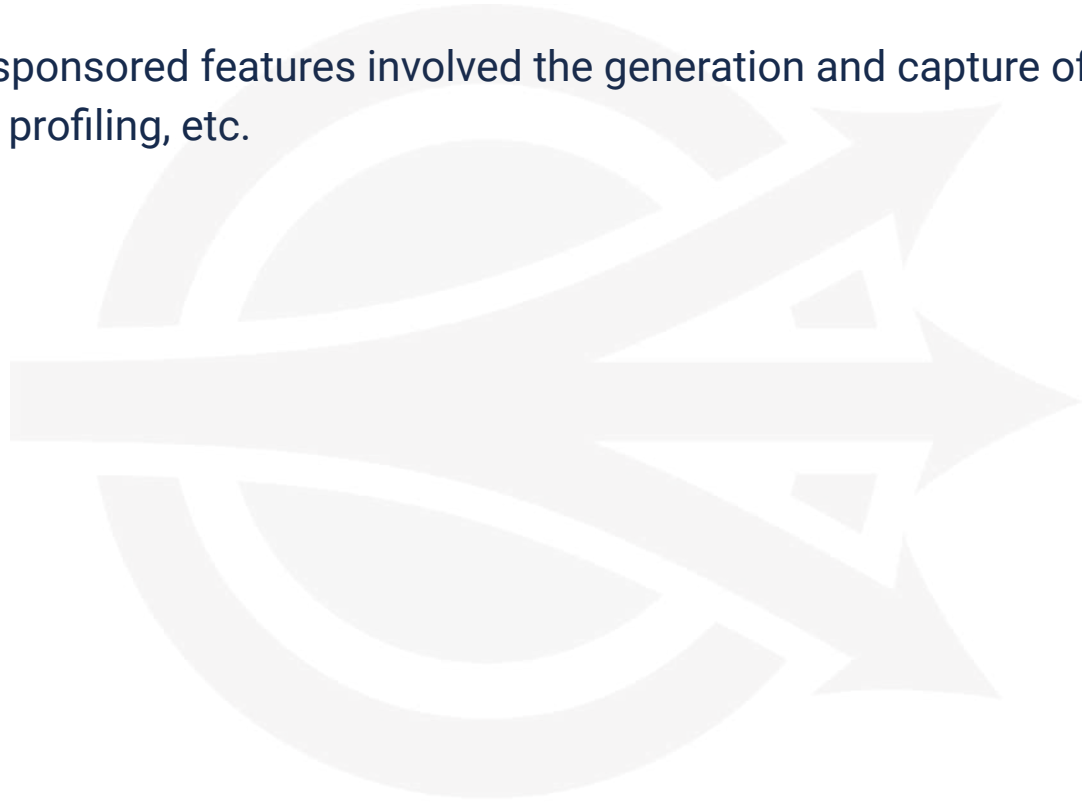
LCOV - code coverage report

Current view:	top level	-DCPS - WinSet.cpp (www.openidds.org)
Test:	coverage-total.info	
Date:	2022-03-30 09:18:32	
Lines:	80	94
Functions:	14	15
		86.1%
		92.3%

```
Line data
1  //
2  //
3  //
4  // distributed under the OpenDDS License.
5  // See: http://www.openidds.org/License.html
6  //
7  //
8  // include "OpenDDS/obj.h" // Only the .obj include should start with open
9  //
10 //
11 //
12 // include "Config.h"
13 //
14 // include "WinSet.h"
15 //
16 // include "WinSetImpl.h"
17 //
18 // namespace
19 //
20 //
21 //
22 //
23 //
24 //
25 //
26 //
27 //
28 //
29 //
30 //
31 //
32 //
33 //
34 //
35 //
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
101 //
102 //
103 //
104 //
105 //
106 //
107 //
108 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 //
132 //
133 //
134 //
135 //
136 //
137 //
138 //
139 //
140 //
141 //
142 //
143 //
144 //
145 //
146 //
147 //
148 //
149 //
150 //
151 //
152 //
153 //
154 //
155 //
156 //
157 //
158 //
159 //
160 //
161 //
162 //
163 //
164 //
165 //
166 //
167 //
168 //
169 //
170 //
171 //
172 //
173 //
174 //
175 //
176 //
177 //
178 //
179 //
180 //
181 //
182 //
183 //
184 //
185 //
186 //
187 //
188 //
189 //
190 //
191 //
192 //
193 //
194 //
195 //
196 //
197 //
198 //
199 //
200 //
201 //
202 //
203 //
204 //
205 //
206 //
207 //
208 //
209 //
210 //
211 //
212 //
213 //
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782 //
783 //
784 //
785 //
786 //
787 //
788 //
789 //
790 //
791 //
792 //
793 //
794 //
795 //
796 //
797 //
798 //
799 //
800 //
801 //
802 //
803 //
804 //
805 //
806 //
807 //
808 //
809 //
810 //
811 //
812 //
813 //
814 //
815 //
816 //
817 //
818 //
819 //
820 //
821 //
822 //
823 //
824 //
825 //
826 //
827 //
828 //
829 //
830 //
831 //
832 //
833 //
834 //
835 //
836 //
837 //
838 //
839 //
840 //
841 //
842 //
843 //
844 //
845 //
846 //
847 //
848 //
849 //
850 //
851 //
852 //
853 //
854 //
855 //
856 //
857 //
858 //
859 //
860 //
861 //
862 //
863 //
864 //
865 //
866 //
867 //
868 //
869 //
870 //
871 //
872 //
873 //
874 //
875 //
876 //
877 //
878 //
879 //
880 //
881 //
882 //
883 //
884 //
885 //
886 //
887 //
888 //
889 //
890 //
891 //
892 //
893 //
894 //
895 //
896 //
897 //
898 //
899 //
900 //
901 //
902 //
903 //
904 //
905 //
906 //
907 //
908 //
909 //
910 //
911 //
912 //
913 //
914 //
915 //
916 //
917 //
918 //
919 //
920 //
921 //
922 //
923 //
924 //
925 //
926 //
927 //
928 //
929 //
930 //
931 //
932 //
933 //
934 //
935 //
936 //
937 //
938 //
939 //
940 //
941 //
942 //
943 //
944 //
945 //
946 //
947 //
948 //
949 //
950 //
951 //
952 //
953 //
954 //
955 //
956 //
957 //
958 //
959 //
960 //
961 //
962 //
963 //
964 //
965 //
966 //
967 //
968 //
969 //
970 //
971 //
972 //
973 //
974 //
975 //
976 //
977 //
978 //
979 //
980 //
981 //
982 //
983 //
984 //
985 //
986 //
987 //
988 //
989 //
990 //
991 //
992 //
993 //
994 //
995 //
996 //
997 //
998 //
999 //
1000 //
```



A number of client-sponsored features involved the generation and capture of meta-data for for logging, debugging, profiling, etc.





The ValueReader and ValueWriter interfaces provide generic support for serializing and deserializing samples.

OpenDDS comes with JsonValueReader and JsonValueWriter for reading/writing samples in JSON.

These have been used mostly for logging (see the RtpsRelay).

However, some users are reading “seed” data stored in JSON to populate test applications.

<https://objectcomputing.com/resources/publications/mnb/logging-samples-as-json-in-opendds-applications>



A basic approach to debugging a DDS application is to log every sample as it is being written and then log every sample as it is read or taken. This leads to a lot of logging code which is difficult to maintain.

The motivation for the Observer interface was to provide a generic capability for monitoring samples at the DCPS layer for events like writing, reading, taking, and storing in the DataReader.

The Observer interface is similar in concept to the spec-defined Listener.

It is possible to log all sample activity with very little code using the Observer with ValueWriter.

<https://objectcomputing.com/resources/publications/mnb/logging-samples-as-json-in-opendds-applications>



The logging in OpenDDS is very “organic” (meaning that lots of different developers over many years did what they thought was right without any guiding principles).

In the new design

- Log levels have been defined for messages intended for users (Error, Warning, Notice, Info)
- Controlled by DCPSLogLevel
- DCPSDebugLevel will be phased out or reinterpreted
- DCPSTransportDebugLevel will be phased out or reinterpreted
- Feature-specific flags will control logging for different features, e.g., security related logging

Log message formatting is being standardized

Log messages will be updated as they are touched in the code.



The ParticipantLocation and ConnectionRecord topics describe the network connectivity (local/relay/ICE) between participants when RTPS discovery and the RTPS transport is used.

Readers for these topics can be accessed like the other built-in topics.

The ParticipantLocation definition is a bit clunky and will eventually be replaced with ConnectionRecord.



Enabling the thread monitoring feature of OpenDDS causes the ThreadStatus built-in topic to be updated.

Each instance corresponds to a thread. If the thread checks in during a monitoring interval, then a sample containing the thread's utilization is written to the topic.

Applications wishing to use this in a watchdog should add a safety factor, e.g., if the status is reported every second, then a problem shouldn't be reported unless there isn't an update in 3 seconds.

For an example, see the RtpsRelay.



OpenDDS has used a variety of techniques to expose data to users (outside of the DDS spec)

- Observer - works well
- Extra BITs - okay, but limited by the way the DDS spec defines BITs
- Polling APIs
- Legacy statistics interface - not maintained
- Transport statistics interface - jury is out

Moving forward

- Move away from polling APIs
- Resolve the weaknesses in BITs but provide the same functionality (sample cache + listener support)



XTypes creates the ability to reflect on types and instances of that type. This is called Dynamic Data.

OpenDDS has partial support for reading Dynamic Data.

The reader received Dynamic Data Values and Dynamic Data Types.

<https://objectcomputing.com/resources/publications/mnb/2022/03/04/reading-data-dynamically-opendds-dynamicdata>



There is a new utility in `/tools/inspect` that uses `DynamicData`.

It takes a topic, type name, and domain id.

It logs samples written to that topic to the console.

We expect to transition from pre-XTypes tools like Recorder/Replayer to XTypes/Dynamic Data tools like inspect.



Goals for OpenDDS v4:

- Improve the initial user experience
- Make it easier to extend and contribute
- Remove superseded features like the udp and multicast transports
- RTPS as default discovery and transport (XTypes, Security, Interoperability)
- Update minimum supported compiler versions, dependencies, etc.
- Continue decoupling from TAO

LET'S CONNECT!

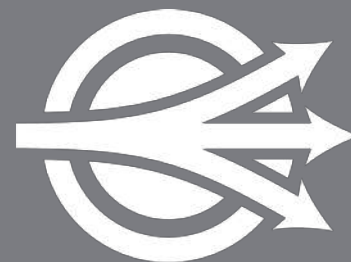
 info@opendds.org

 opendds.org/foundation

 [@OpenDDS](https://twitter.com/OpenDDS)

 linkedin.com/showcase/opendds

 github.com/objectcomputing/OpenDDS



OpenDDS[®]

Thank you!
Any Questions?



OpenDDS
FOUNDATION™

