GSFC JPSS CMO
November 11, 2013
Released

Effective Date: November 7, 2013
Block/Revision 0123B

Joint Polar Satellite System (JPSS) Ground Project
Code 474
474-00019-02-B0123

Joint Polar Satellite System (JPSS) Application Programming Interface (API) User's Guide, Volume II - Java and Java Messaging Service (JMS)

For Public Release

The information provided herein does not contain technical data as defined in the International Traffic in Arms Regulations (ITAR) 22 CFC 120.10. This document has been approved For Public Release to the NOAA Comprehensive Large Array-data Stewardship System (CLASS).

Block 1.2.3

Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics and
Space Administration

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Joint Polar Satellite System (JPSS) Application Programming Interface (API) User's Guide, Volume II – Java and Java Messaging Service (JMS)

JPSS Electronic Signature Page

Prepared By:

Thomas Jennings
JPSS Ground Project System Engineer
(Electronic Approvals available online at https://jpssmis.gsfc.nasa.gov/mainmenu_dsp.cfm)

Reviewed By:

Leslye Boyce
JPSS Ground Project Systems Manager
(Electronic Approvals available online at https://jpssmis.gsfc.nasa.gov/mainmenu_dsp.cfm)

Goddard Space Flight Center
Greenbelt, Maryland

Preface

This document is under JPSS Ground ERB configuration control. Once this document is approved, JPSS approved changes are handled in accordance with Class I and Class II change control requirements as described in the JPSS Configuration Management Procedures, and changes to this document shall be made by complete revision.

Any questions should be addressed to:

JPSS Configuration Management Office
NASA/GSFC
Code 474
Greenbelt, MD 20771

Change History Log

Revision	Effective Date	Description of Changes (Reference the CCR & CCB/ERB Approval Date; for first Block Version Release, identify origin of document source)	Sections Affected
0123-	Aug 23, 2012	This version incorporates Rev- of 474-00019-02-B0122, dated January 26, 2012, to create the baseline for Block 1.2.3, Rev-. This was approved (out of board) by the JPSS Ground ERB via 474-CCR-12-0512 on the effective date shown.	All
0123A	Mar 21, 2013	This version incorporates 474-CCR-13-0899 (ECR-CGS-0176). This was approved by the JPSS Ground ERB on the effective date shown.	2.2.1.3 2.2.4.1.14 2.2.6 2.2.6.1 2.2.7 2.2.7.1 2.2.8.1 2.2.15.1 2.2.21 2.2.26 2.2.28 2.2.31 2.2.32 2.2.33
0123A-1	Oct 24, 2013	This version incorporates 474-CCR-13-1216 (ECR-CGS-0209) which was approved by the JPSS Ground ERB on the effective date shown	2.2.1.3.53 2.2.1.3.54 2.2.5.1.10 2.2.9 Figure 2.2.9-1 2.2.9.1 2.2.9.1.1 2.2.9.1.2 2.2.9.1.3 2.2.9.1.4 2.2.15.1.9 2.2.15.1.16 2.2.15.1.28 2.2.33
0123B	Nov 07, 2013	This version incorporates 0123A-1 and 474-CCR-13-0998. This was approved by the JPSS Ground ERB on the effective date shown.	All

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Northrop Grumman Space & Mission Systems Corp. Space Technology

One Space Park
Redondo Beach, CA 90278

Engineering and Manufacturing Development (EMD)
Acquisition & Operations Contract

CAGE NO. 11982

National Polar-Orbiting Operational Environmental Satellite System (NPOESS) Application Programming Interface (API) User's Guide, Volume II – Java and Java Messaging Service (JMS)

DOCUMENT DATE: 08/19/2009 **DOCUMENT NO. D41044-02**

CDRL A009 **REV. C**

POINT OF CONTACT: Terri Matthews, Systems Integration

ELECTRONIC APPROVAL SIGNATURES:

Clark Snodgrass, SEITO Director

Fabrizio Pela, SEIT IPT Lead

Keith Reinke, Ground Segments IPT Lead

Mary Ann Chory, Space Segment IPT Lead

Ben James, Operations and Support IPT Lead

David Vandervoet, NPOESS Program Manager

Prepared by
Northrop Grumman Space Technology
One Space Park
Redondo Beach, CA 90278

Under
Contract No. F04701-02-C-0502

Prepared for
Department of the Air Force
NPOESS Integrated Program Office
C/O SMC/CIK
2420 Vela Way, Suite 1467-A8
Los Angeles AFB, CA 90245-4659

DISTRIBUTION STATEMENT F: Distribution statement "F" signifies that further dissemination should only be made as directed by the controlling DoD Office (NPOESS IPO). Ref DODD 5230.24.

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Northrop Grumman Space & Mission Systems Corp. Space Technology
One Space Park
Redondo Beach, CA 90278

Revision/Change Record

For Document No. D41044-02

Revision	Document Date	Revision/Change Description	Pages Affected
---	03/09/2007	Initial release. Divided into two volumes from the basic document with reorganization of content. Incorporated ER 560	All
A	06/20/2008	Full revision of the document to incorporate IDPS B1.5 API documentation. This revision of the document changes the content and structure of the API for the C++, Java, and JMS APIs The Revision, ECR 778, does not incorporate any additional DCOs	All
B	01/23/09	ECR 891A is a revision of the document to incorporate comments from CCR and CDA reviews. • Clarified the use of IET in several places and specified the precision of times in several places • Corrected definitions of Latitude in several places as well as the Latitude Values in several of the examples • Miscelaneous grammar changes for clarity • Deleted Section 2.2.3 DDSXML_DataProduct Class Reference • Corrected the return type for several methods	All
C	08/19/09	Revision A of ECR 985 removes comments and updates Section 2.2.1.3.28 ,Section 2.2.7.1.4, Section 2.2.28.2.7, Section 2.2.30.1.1, and Section 2.2.30.1.2 ECR 985 incorporates updates to bring it into alignment with IDPS Build 1.5.X.1. Includes updates to compilers and COTS used for the API. Also incorporates ECR 954, Deletion of C API	All

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Table of Contents

1. INTRODUCTION 1
1.1 Document Purpose 1
1.2 Document Overview 1
2. JAVA API Documentation 2
2.1 Coding Conventions 2
2.1.1 JAVA Coding Conventions 2
2.1.2 Environment Variables 3
2.1.3 Procedures for Client-side DDS API SSL Certificate Installation 4
2.2 JAVA API Module Documentation List 5
2.2.1 DDSAPI_Message Class Reference 5
2.2.2 DDSXML_CatalogRequest Class Reference 27
2.2.3 Deleted 29
2.2.4 DDSXML_DataProductList Class Reference 29
2.2.5 DDSXML_DataShipment Class Reference 36
2.2.6 DDSXML_DataShipmentList Class Reference 40
2.2.7 DDSXML_Destination Class Reference 45
2.2.8 DDSXML_DestinationList Class Reference 54
2.2.9 Deleted 57
2.2.10 DDSXML_GEORequest Class Reference 57
2.2.11 DDSXML_ImplementationRequestTypesEnum Class Reference 70
2.2.12 DDSXML_PeriodicRequest Class Reference 74
2.2.13 DDSXML_ProductRequest Class Reference 77
2.2.14 DDSXML_QueryRequest Class Reference 83
2.2.15 DDSXML_Request Class Reference 85
2.2.16 DDSXML_RequestTypesEnum Class Reference 92
2.2.17 DDSXML_StandardRequest Class Reference 95
2.2.18 DDSXML_SystemMessage Class Reference 96
2.2.19 DDSXML_SystemMessageList Class Reference 99
2.2.20 DDSXML_TemporalRequest Class Reference 100
2.2.21 DDSXML_User Class Reference 104
2.2.22 DDSXML_DataShipmentStatesEnum Class Reference 106
2.2.23 DDSXML_SystemMessageSeverityEnum Class Reference 108
2.2.24 DDSXML_DestinationTransferTypeEnum Class Reference 111
2.2.25 DDSXML_DestinationStatesEnum Class Reference 113
2.2.26 DDSXML_UserList Class Reference 115
2.2.27 DDSXML_ClientTypesEnum Class Reference 118
2.2.28 DDSXML_DataProductIDList Class Reference 121
2.2.29 DDSXML_Longitude Class Reference 126
2.2.30 DDSXML_Latitude Class Reference 128
2.2.31 DDSXML_ResultSet Class Reference 130
2.2.32 DDSXML_ResultSetList Class Reference 132
2.2.33 DDSXML_DestinationEntry Class Reference 135
3. JAVA JMS Documentation 142
3.1 Coding Conventions 143

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

3.2 Java JMS Module Documentation List 143
3.2.1 Environment Variables 143
3.3 Java JMS API Module Documentation List 144
3.3.1 DDSXML_Login Class Reference 144
3.3.2 DDSXML_LoginResponse Class Reference 146
3.3.3 DDSXML_Logout Class Reference 148
3.3.4 DDSXML_LogoutResponse Class Reference 150
3.3.5 DDSXML_CreateRequest Class Reference 152
3.3.6 DDSXML_CreateRequestResponse Class Reference 154
3.3.7 DDSXML_ModifyRequest Class Reference 156
3.3.8 DDSXML_ModifyRequestResponse Class Reference 158
3.3.9 DDSXML_DeleteRequest Class Reference 160
3.3.10 DDSXML_DeleteRequestResponse Class Reference 162
3.3.11 DDSXML_SuspendRequest Class Reference 164
3.3.12 DDSXML_SuspendRequestResponse Class Reference 166
3.3.13 DDSXML_ResumeRequest Class Reference 168
3.3.14 DDSXML_ResumeRequestResponse Class Reference 170
3.3.15 DDSXML_GetRequestByID Class Reference 172
3.3.16 DDSXML_GetRequestByIDResponse Class Reference 174
3.3.17 DDSXML_JMSRetransmit Class Reference 176
3.3.18 DDSXML_JMSRetransmitResponse Class Reference 178
3.3.19 DDSXML_CreateDestination Class Reference 180
3.3.20 DDSXML_CreateDestinationResponse Class Reference 182
3.3.21 DDSXML_GetDestinations Class Reference 184
3.3.22 DDSXML_GetDestinations Class Reference 186
3.3.23 DDSXML_GetDataProducts Class Reference 188
3.3.24 DDSXML_GetDataProductsResponse Class Reference 190
3.3.25 DDSXML_GetDataShipmentList Class Reference 192
3.3.26 DDSXML_JMSSubscribeToSystemMessages Class Reference 194
3.3.27 DDSXML_JMSSubscribeToSystemMessagesResponse Class Reference 196
3.3.28 DDSXML_JMSStopSubscribing Class Reference 198
3.3.29 DDSXML_JMSStopSubscribingResponse Class Reference 200
3.3.30 DDSXML_GetRequestIDs Class Reference 202
3.3.31 DDSXML_GetRequestIDsResponse Class Reference 203
3.3.32 DDSXML_DeleteDestinations Class Reference 203
3.3.33 DDSXML_DeleteDestinationsResponse Class Reference 205
3.3.34 DDSXML_GetStatusMessages Class Reference 206
3.3.35 DDSXML_ GetStatusMessageResponse Class Reference 207
3.3.36 DDSXML_GetDataShipmentListResponse Class Reference 209
3.3.37 DDSXML_CommandsEnum Class Reference 210
3.3.38 DDSXML_CommandStateEnum Class Reference 235
3.3.39 DDSXML_ClientTypesEnum Class Reference 238
3.3.40 DDSXML_User Class Reference 240
3.3.41 DDSXML_Data Class Reference 240
Appendix A System Requirements 248
Appendix B Document Specific Acronyms List 249

List of Figures

Figure 2.2.1-1, DDSAPI_Message Class UML Diagram 6
Figure 2.2.2-1, DDSXML_CatalogRequest Class UML Diagram 28
Figure 2.2.4-1, DDSXML_DataProductList Class UML Diagram 29
Figure 2.2.5-1, DDSXML_DataShipment Class UML Diagram 36
Figure 2.2.6-1, DDSXML_DataShipmentList Class UML Diagram 41
Figure 2.2.7-1, DDSXML_Destination Class UML Diagram 46
Figure 2.2.8-1, DDSXML_DestinationList Class UML Diagram 54
Figure 2.2.10-1, DDSXML_GEORequest Class UML Diagram 58
Figure 2.2.11-1, DDSXML_ImplementationRequestTypesEnum Class UML Diagram 71
Figure 2.2.12-1, DDSXML_PeriodicRequest Class UML Diagram 74
Figure 2.2.13-1, DDSXML_ProductRequest Class UML Diagram 78
Figure 2.2.14-1, DDSXML_QueryRequest Class UML Diagram 84
Figure 2.2.15-1, DDSXML_Request Class UML Diagram 86
Figure 2.2.16-1, DDSXML_RequestTypesEnum Class UML Diagram 92
Figure 2.2.17-1, DDSXML_StandardRequest Class UML Diagram 96
Figure 2.2.18-1, DDSXML_SystemMessage Class UML Diagram 97
Figure 2.2.19-1, DDSXML_SystemMessageList Class UML Diagram 99
Figure 2.2.20-1, DDSXML_TemporalRequest Class UML Diagram 101
Figure 2.2.21-1, DDSXML_User Class UML Diagram 104
Figure 2.2.22-1, DDSXML_DataShipmentStatesEnum Class UML Diagram 106
Figure 2.2.23-1, DDSXML_ SystemMessageSeverityEnum Class UML Diagram 109
Figure 2.2.24-1, DDSXML_ DestinationTransferTypeEnum Class UML Diagram 111
Figure 2.2.25-1, DDSXML_ DestinationStatesEnum Class UML Diagram 114
Figure 2.2.26-1, DDSXML_ UserList Class UML Diagram 116
Figure 2.2.27-1, DDSXML_ ClientTypesEnum Class UML Diagram 119
Figure 2.2.28-1, DDSXML_ DataProductIDList Class UML Diagram 122
Figure 2.2.29-1, DDSXML_ Longitude Class UML Diagram 126
Figure 2.2.30-1, DDSXML_ Latitude Class UML Diagram 129
Figure 2.2.31-1, DDSXML_ ResultSet Class UML Diagram 130
Figure 2.2.32-1, DDSXML_ ResultSetList Class UML Diagram 133
Figure 3.0-1, IDPS JMS Interface Graphical Depiction 142
Figure 3.3.1-1, DDSXML_Login Class UML Diagram 145
Figure 3.3.2-1, DDSXML_LoginResponse Class UML Diagram 147
Figure 3.3.3-1, DDSXML_Logout Class UML Diagram 149
Figure 3.3.4-1, DDSXML_LogoutResponse Class UML Diagram 151
Figure 3.3.5-1, DDSXML_CreateRequest Class UML Diagram 153
Figure 3.3.6-1, DDSXML_ CreateRequestResponse Class UML Diagram 155
Figure 3.3.7-1, DDSXML_ModifyRequest Class UML Diagram 157
Figure 3.3.8-1, DDSXML_ModifyRequestResponse Class UML Diagram 159
Figure 3.3.9-1, DDSXML_DeleteRequest Class UML Diagram 161
Figure 3.3.10-1, DDSXML_DeleteRequestResponse Class UML Diagram 163
Figure 3.3.11-1, DDSXML_SuspendRequest Class UML Diagram 165
Figure 3.3.12-1, DDSXML_SuspendRequestResponse Class UML Diagram 167
Figure 3.3.13-1, DDSXML_ResumeRequest Class UML Diagram 169

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Figure 3.3.14-1, DDSXML_ResumeRequestResponse Class UML Diagram 171
Figure 3.3.15-1, DDSXML_GetRequestByID Class UML Diagram..... 173
Figure 3.3.16-1, DDSXML_GetRequestByIDResponse Class UML Diagram 175
Figure 3.3.17-1, DDSXML_JMSRetransmit Class UML Diagram 177
Figure 3.3.18-1, DDSXML_JMSRetransmitResponse Class UML Diagram 179
Figure 3.3.19-1, DDSXML_CreateDestination Class UML Diagram 181
Figure 3.3.20-1, DDSXML_CreateDestinationResponse Class UML Diagram 183
Figure 3.3.21-1, DDSXML_GetDestinations Class UML Diagram..... 185
Figure 3.3.22-1, DDSXML_GetDestinationsResponse Class UML Diagram 187
Figure 3.3.23-1, DDSXML_GetDataProducts Class UML Diagram 189
Figure 3.3.24-1, DDSXML_GetDataProductsResponse Class UML Diagram..... 191
Figure 3.3.25-1, DDSXML_GetDataShipmentList Class UML Diagram 193
Figure 3.3.26-1, DDSXML_JMSSubscribeToSystemMessages Class UML Diagram 195
Figure 3.3.27-1, DDSXML_JMSSubscribeToSystemMessagesResponse Class UML Diagram 197
Figure 3.3.28-1, DDSXML_JMSStopSubscribing Class UML Diagram 199
Figure 3.3.29-1, DDSXML_JMSStopSubscribingResponse Class UML Diagram 201
Figure 3.3.30-1, DDSXML_GetRequestIDs Class UML Diagram 202
Figure 3.3.31-1, DDSXML_GetRequestIDsResponse Class UML Diagram..... 203
Figure 3.3.32-1, DDSXML_DeleteDestinations Class UML Diagram 204
Figure 3.3.33-1, DDSXML_DeleteDestinationsResponse Class UML Diagram..... 205
Figure 3.3.34-1, DDSXML_GetStatusMessages Class UML Diagram 207
Figure 3.3.35-1, DDSXML_ GetStatusMessageResponse Class UML Diagram 208
Figure 3.3.36-1, DDSXML_GetDataShipmentListResponse Class UML Diagram 209
Figure 3.3.37-1, DDSXML_CommandsEnum Class UML Diagram..... 211
Figure 3.3.38-1, DDSXML_CommandStateEnum Class UML Diagram 236
Figure 3.3.39-1, DDSXML_ClientTypesEnum Class UML Diagram 238
Figure 3.3.41-1, DDSXML_Data Class UML Diagram..... 241

List of Tables

Table 2.1-1, Java Coding Conventions 2
Table B-1, Document-Specific Acronym List 249

1. INTRODUCTION

1.1 Document Purpose

This is the second volume of a two volume document. General information about the NPOESS Application Programming Interface (API) User's Guide and how to make requests for data from the system are found in the Joint Polar Satellite System (JPSS) Application Programming Interface (API) User's Guide Volume I - C++, 474-00019-01. In particular, the details on the various types of requests and the definitions of the request parameters can be found in that volume. This volume only contains the code signatures for the Java and Java Messaging Services (JMS) APIs.

1.2 Document Overview

Section 1 Introduction – This section provides introductory material for Volume II of this document as an accompaniment to the material provided in the NPOESS API User's Guide Volume I.

Section 2 Java API – This section provides the information for the Java version of the API.

Section 3 JMS API – This section provides the information for the JMS version of the API.

Appendix A System Requirements – This section contains information pertaining to the software needs of the NPOESS API.

Appendix B Document Specific Acronyms List – Provides a list of acronyms unique to this document. All other acronyms are identified and listed in the NPOESS Acronyms, D35838.

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

2. JAVA API DOCUMENTATION

The Java API is a set of libraries that can be used with any Java 1.4 JVM on either a Microsoft® Windows® or IBM AIX® platform. The user application is defined to be the application that is using the API. The central object of the API is the Message object (in the dds.RequestAPI.Request package). All NPOESS API classes are in the RequestAPI.jar file. There should only be one Message object instantiated for the user application.

The default constructor should be used to construct the Message class. The default constructor assumes that the variables DDS_ PORT and DDS_ HOST are instantiated as follows for Unix:

-DDDS_PORT=$NPOESS_DPE_DDS_PORT

-DDDS_HOST=$NPOESS_DPE_DDS_HOST

and as follows for windows:

-DDDS_HOST="% NPOESS_DPE_DDS_HOST%"

-DDDS_PORT="% NPOESS_DPE_DDS_PORT%"

The NPOESS_DPE_DDS_PORT and NPOESS_DPE_DDS_HOST are specified in the environment and may be defined during installation or instantiation.

2.1 Coding Conventions

The coding conventions used for the NPOESS API comply with the NPOESS Software Standard and Practices Manual, MN60822-PMO-001.

2.1.1 JAVA Coding Conventions

All Java code follows the conventions in Table 2.1-1, Java Coding Conventions.

Table 2.1-1, Java Coding Conventions

Java Coding Conventions	
Application	**Convention**
Constants	All constants meant for users of the class are static and are publicly accessible.
Non-static variables	Non-static variables are private and may only be accessed through setters and getters. If a setter or getter doesn't exist, then that variable cannot be accessed in that fashion.
Default constructors	All classes have a default constructor. Some of the default constructors may not be accessible to the user application, since those classes should not be instantiated in that manner.

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Java Coding Conventions	
Application	**Convention**
Error exception reporting	All errors are reported as a Java exception derived from the Exception class. There are four types of exceptions that may be thrown: • RequestException • MessageException • ValidationException • DDSException DDSException is the base exception class for the other three, so catching the DDSException is sufficient when calling methods that throw any of these four exceptions.
toString()	The toString() is overridden in classes where necessary. This is done to facilitate the printing out of important contents of the particular object referenced.
equals() method	The equals() method exists in classes that can be compared. In particular, for data product and request related data.
Serializing Objects	Certain objects may be serialized. Those objects implement the java.io.Serializable interface. This interface is for data product and request related data.
Cloning Objects	Certain objects may be cloned. Those objects implement the java.lang.Cloneable interface. These are for data product and request related data.
Memory Management	All requests, templates, data products, and catalogs are managed through the Message object. **These objects will be created and destroyed by the API. Do not attempt to manage these objects in your code.** The Message object needs to be instantiated and deleted by the user application.

Refer to the NPOESS Software Standards and Practices Manual (SSPM), MN60822-PMO-001, Appendix F, for additional details on coding guidelines.

2.1.2 Environment Variables

Following is a list of the environment variables that need to be specified for use by the Java Client.

The **DDS_PORT** environment variable tells the Java API application what port to use when communicating with the DDS Server.

The **DDS_HOST** environment variable tells the Java API application the IP address of the machine that is hosting the DDS Server.

The **DDS_TIMEOUT** variable tells the Java API how many minutes of inactivity before closing the session.

The **DDS_ROOT** variable identifies the location where IDPS log files should be stored if necessary.

The **DINFUTIL_CFGDOMAIN** environment variable tells the Java API application the location of the "INF_GuideList.cfg" file.

The **DHTTPS_PROTOCOL** variable tells the Java API application if

HTTPS_PROTOCOL is either http or https

The **DHTTPS_HOST** variable tells the Java API application the IP address of the host that is running the Apache Web Server

The **DDS_HEARTBEAT_DURATION** variable tells the Java API application the duration in seconds when the Server will time out and disconnect the API if the heartbeat is not received.

The **VBROKERDIR** variable identifies the location of Visibroker COTS JAR files (lib\vbjorb.jar, lib\vbjdev.jar, lib\lm.jar, lib\vbsec.jar).

Notes:

The IDPS Windows Installshield creates this environment variable when it's installed

2.1.3 Procedures for Client-side DDS API SSL Certificate Installation

Overview:

- First obtain the updated certificates from the (Mission Operations Team) MOT Configuration Manager (CM).
- Add the renewed certificates to the keystore.

Note:
Renewed certificate name will contain the site, domain and renewal date so that multiple versions may be stored in case of a rollback.
Example: **myca.nesdis.ops.2010Feb14.crt**

AIX/Unix/Linux Java Client Import Procedures:
Copy or FTP the certificate to a working directory of the target environment.

Example: **/tmp/working_dir/myca.nesdis.ops.2010Feb14.crt**

Import the certificate into the keystore.

> keytool -import -alias <key> -keystore cacerts -file <certificate>
Example:
> /usr/java6_64/bin/keytool -noprompt -import -alias mykey.`hostname`.`date +"%Y%m%d"` -keystore /usr/java6_64/jre/lib/security/cacerts -file /tmp/working_dir/myca.nesdis.ops.2010Feb14.crt

Repeat the steps for all servers that will connect to IDPS via the DDS API.

Windows Java Client Import Procedures:
Copy or FTP the certificate to a working directory of the target environment.

Example: **E:\temp\working_dir\myca.nesdis.ops.2010Feb14.crt**

Import the certificate into the keystore.

E:\> keytool -import -alias mykey -file <certificate> -keystore \cacerts
Example:
E:\> E:\npoess\apps\jdk1.6.0_17\bin\keytool -import -alias mykey -file E:\temp\working_dir\myca.nesdis.ops.2010Feb14.crt -keystore E:\npoess\apps\jdk1.6.0_17\jre\lib\security\cacerts

Repeat the steps for all servers that will connect to IDPS via the DDS API.

2.2 JAVA API Module Documentation List

The Java API consists of a set of classes defining the attributes, enumerations, and functions that allow the user to logon, create a request, process a request, and perform the basic manipulations on catalog items and templates.

2.2.1 DDSAPI_Message Class Reference

This object is responsible for establishing and maintaining contact with the API Manager. This class is also responsible for handling commands that can be performed in the system. The API commands are executed as calls on methods in this class. Most pointers return a copy of memory referenced by the API. It should be deleted by the caller. The caller keeps ownership to the pointers passed in. The API does not delete the data passed in. A user of the API must create an instance of this class to interact with the API. All interaction with the API should be done through this object or objects returned by this object. After this class is created a user of this class must login to the API to use it. Calls may then be made on public methods in this class or in the classes returned by these methods. If there are problems executing the methods in the API then a message will be created and added to the System Messages. A user can then use the getSystemMessages() call to get the current system messages. If there is a problem with the API use of a method, that method will exit with either a false, a null string, or an empty vector. Under normal API operation an exception should not be passed back to the caller of any API method. Do not try to create objects outside of this class. The API will not know about them or be able to use them.

The Class diagram representing the DDSAPI_Message Class is provided in Figure 2.2.1-1, DDSAPI_Message Class UML Diagram.

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

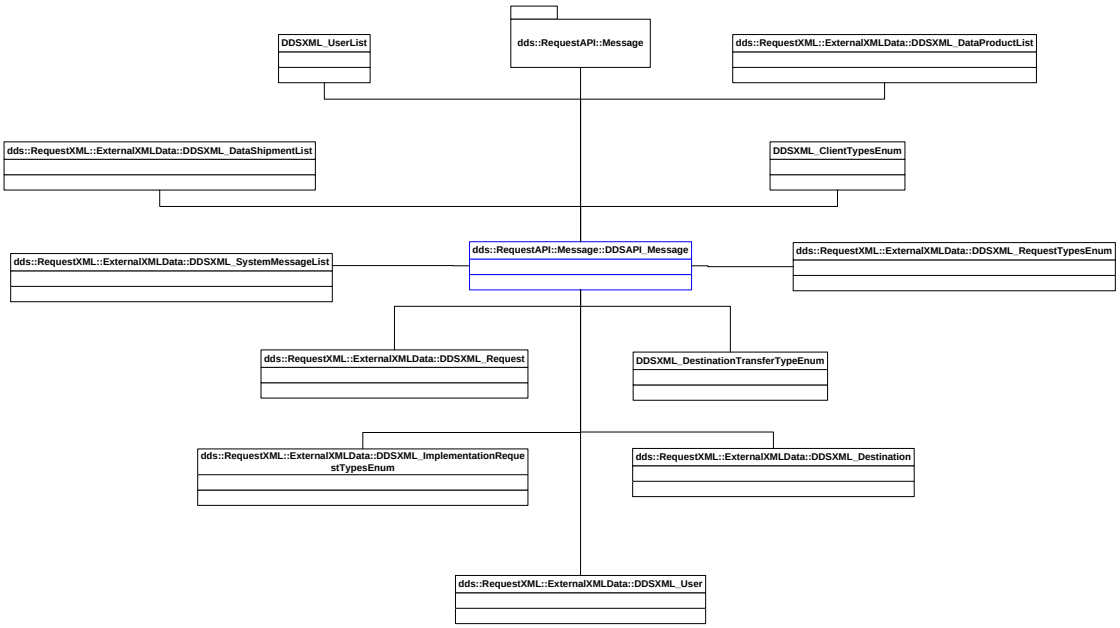

Figure 2.2.1-1, DDSAPI_Message Class UML Diagram

2.2.1.1 DDSAPI_Message Class Attributes

- static final String dds::RequestAPI::Message::DDSAPI_Message::DEFAULT_HOST = "NPOESS_DPE_DDS_API_DEFAULT_HOST" – The default environmental variable that stores the host name to connect to.
- static final String dds::RequestAPI::Message::DDSAPI_Message::DEFAULT_PORT = "NPOESS_DPE_DDS_API_DEFAULT_PORT" – The default environmental variable that stores the port number to connect to.
- static final int dds::RequestAPI::Message::DDSAPI_Message::DEFAULT_TIMEOUT = 60 – The default time out for the Message.

2.2.1.2 DDSAPI_Message Class Constructors

2.2.1.2.1 DDSAPI_Message::DDSAPI_Message

dds::RequestAPI::Message::DDSAPI_Message::DDSAPI_Message (

String hostname,

String port,

int timeout,

DDSXML_ClientTypesEnum clientType,

boolean secureFlag

)

Overloaded Constructor. This constructor must be used to create the DDS API interface to the DDS Server. The default constructor should not be used. The proper host and port must be used that matches the DDS Server setup.These are configurable since they may change for different Server connections. This should be the only case in this object where an exception will be thrown since we can not create an instance of this object.

Parameters:

- hostname The environmental variable to retrieve the host name of the Request Server from.
- port The environmental variable to retrieve the port number from.
- timeout The timeout duration in seconds.
- clientType The client type to create - CLIENT_API
- secureFlag - True to use https

2.2.1.2.2 DDSAPI_Message::DDSAPI_Message

dds::RequestAPI::Message::DDSAPI_Message::DDSAPI_Message (

String hostname,

String port,

int timeout,

DDSXML_ClientTypesEnum clientType

)

Overloaded Constructor. This constructor must be used to create the DDS API interface to the DDS Server. The default constructor should not be used. The proper host and port must be used that matches the DDS Server setup. Make sure that these are configurable as they may change for different Server connections. This should be the only case in this object where an exception will be thrown since we can not create an instance of this object.

Parameters:

- hostname The environmental variable to retrieve the host name of the Request Server from.
- port The environmental variable to retrieve the port number from.
- timeout The timeout duration in seconds.
- clientType The client type to create – CLIENT_API

2.2.1.3 DDSAPI_Message Class Functions

2.2.1.3.1 DDSAPI_Message::login

boolean dds::RequestAPI::Message::DDSAPI_Message::login (

String username,

String password,

String role

)

This method provides a mechanism for user login into the system. The user must provide a username, and password. The role parameter is optional if this user only has one role. If authentication fails, false will be returned to the caller. The API must also be configured to use the API calls.

Parameters:

- username The username to be used to authenticate the user.
- password The password associated with the username.
- role The role that is associated with the username for this login.

Returns:

- boolean true - if the user has successfully logged into the system. false - if the user login attempt has failed.

2.2.1.3.2 DDSAPI_Message::logout

boolean dds::RequestAPI::Message::DDSAPI_Message::logout (

)

Sends a message to the server that the API is ready to log out the user. The session is only disconnected on timeout or if this object is deleted. This must be called when you are done using the API. The Server uses this to clean up internal memory faster.

Returns:

- boolean true - if the logout was successful false - if the logout attempt failed

2.2.1.3.3 DDSAPI_Message::getLoginState

boolean dds::RequestAPI::Message::DDSAPI_Message::getLoginState (

)

This method returns the login state of the system. Make sure to look at the configuration state also.

Returns:

- boolean The login state, true - logged in false - no logged in

2.2.1.3.4 DDSAPI_Message::getConfigState

boolean dds::RequestAPI::Message::DDSAPI_Message::getConfigState (

)

This method returns the config state of the system. If true then the API has been fully configured. If false then a part or all of the API has not been configured. The API must be fully configured to use the other methods in the API.

Returns:

- boolean The config state, true - configured false - not configured

2.2.1.3.5 DDSAPI_Message::getRoles

Vector dds::RequestAPI::Message::DDSAPI_Message::getRoles (

String username,

String password

)

This method returns a vector of strings that contain all of the user's valid roles as strings. This is the only API method that does not require the user to be logged in before it can be called. This is to allow a display/GUI to show a user all of their roles when logging into the DDS Server.

Parameters:

- username The user's username whose roles are to be obtained from the Request Server.
- password The user's password whose roles are to be obtained from the Request Server.

Returns:

- Vector The vector of strings that contain the user's possible roles.

2.2.1.3.6 DDSAPI_Message::getSystemMessages

Vector dds::RequestAPI::Message::DDSAPI_Message::getSystemMessages (

DDSXML_User subUser

)

This method returns a vector of DDSXML_SystemMessage pointers to the user. The pointers point to memory that is owned by the caller and it is the responsibility of the user to free this memory.

Parameters:

- subUser The user that is requesting this API call.

Returns:

- Vector The vector of DDSAPI_SystemMessages received by the DDSAPI_Message.

2.2.1.3.7 DDSAPI_Message::getStoredSystemMessages

Vector dds::RequestAPI::Message::DDSAPI_Message::getStoredSystemMessages (

DDSXML_User subUser

)

subUser This method returns a vector of DDSXML_SystemMessage pointers to the user. The pointers point to memory that is owned by the caller and it is the responsibility of the user to free this memory.

Parameters:

- subUser The user that is requesting this API call.

Returns:

- Vector The vector of DDSAPI_SystemMessages received by the DDSAPI_Message.

2.2.1.3.8 DDSAPI_Message::getDataShipments

Vector dds::RequestAPI::Message::DDSAPI_Message::getDataShipments (

String requestID,

DDSXML_User user

)

This method returns a vector of DDSXML_DataShipment pointers to the user.

Parameters:

- requestID The ID of the DDSXML_Request to get the shipment records for.
- user The user that is requesting this API call.

Returns:

- Vector The vector of DDSXML_DataShipment received by the API.

Deprecated:

Use getDataShipmentList

2.2.1.3.9 DDSAPI_Message::addDestination

boolean dds::RequestAPI::Message::DDSAPI_Message::addDestination (

String destinationName,

String hostName,

String path,

String ftpUserName,

String ftpPassword,

DDSXML_DestinationTransferTypeEnum transferType,

DDSXML_User subUser

)

This method adds a new destination to the user defined destination list.

Parameters:

- destinationName The name for this destination.
- hostName The hostname or IP address.
- path The destination path.
- ftpUserName The username for the destination FTP server.
- ftpPassword The password for the destination FTP server.
- transferType The transferType for the destination.
- subUser The user that is requesting this API call.

Returns:

- boolean True if added OK to the Server, false if not.

2.2.1.3.10 DDSAPI_Message::addDestinationReturnResult

```
boolean dds::RequestAPI::Message::DDSAPI_Message::addDestinationReturnResult (
    String destinationName,
    String hostName,
    String path,
    String ftpUserName,
    String ftpPassword,
    DDSXML_DestinationTransferTypeEnum transferType,
    DDSXML_User subUser
  )
```

This method adds a new destination to the user defined destination list.

Parameters:

- destinationName The name for this destination.
- hostName The hostname or IP address.
- path The destination path.
- ftpUserName The username for the destination FTP server.
- ftpPassword The password for the destination FTP server.
- transferType The transferType for the destination.
- subUser The user that is requesting this API call.

Returns:

boolean True if added OK to the Server, false if not.

Deprecated:

Use addDestination

2.2.1.3.11 DDSAPI_Message::addNewDestination

String dds::RequestAPI::Message::DDSAPI_Message::addNewDestination (

String destinationName,

String hostName,

String path,

String ftpUserName,

String ftpPassword,

DDSXML_DestinationTransferTypeEnum transferType,

DDSXML_User subUser

)

This method adds a new destination to the user defined destination list.

Parameters:

- destinationName The name for this destination.
- hostName The hostname or IP address.
- path The destination path.
- ftpUserName The username for the destination FTP server. Not required for a local destination
- ftpPassword The password for the destination FTP server. Not required for a local destination
- transferType The transferType for the destination.
- subUser The user that is requesting this API call.

Returns:

- String Destination ID or "" if there was an error.

2.2.1.3.12 DDSAPI_Message::deleteDestination

boolean dds::RequestAPI::Message::DDSAPI_Message::deleteDestination (

DDSXML_Destination destination,

DDSXML_User subUser

)

This method removes a user defined destination from the user destination list.

Parameters:

- destination The destination to be removed from the destination list.

- subUser The user that is requesting this API call.

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.13 DDSAPI_Message::getDestinations

Vector dds::RequestAPI::Message::DDSAPI_Message::getDestinations (

DDSXML_User subUser

)

This method returns all this users defined destinations in a vector to the caller.

Parameters:

- subUser The user that is requesting this API call.

Returns:

- Vector The vector containing all user-defined destinations.

2.2.1.3.14 DDSAPI_Message::getDestination

DDSXML_Destination dds::RequestAPI::Message::DDSAPI_Message::getDestination (

String userIndex,

DDSXML_User subUser

)

This method returns the user identified by the destination ID if it exists in the system.

Parameters:

- userIndex The userIndex to check for
- subUser The user that is requesting this API call.

Returns:

- DDSXML_Destination A pointer to the destination . 0 - The destination was not found. Valid pointer otherwise.

2.2.1.3.15 DDSAPI_Message::modifyDestination

boolean dds::RequestAPI::Message::DDSAPI_Message::modifyDestination (

DDSXML_Destination destination,

DDSXML_User subUser

)

This method modifies the user destination.

Parameters:

- destination The user defined destination to be modified.

- subUser The user of this command

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.16 DDSAPI_Message::getDataProductList

```
DDSXML_DataProductList
dds::RequestAPI::Message::DDSAPI_Message::getDataProductList (
    DDSXML_User subUser
  )
```

This method returns a DDSXML_DataProductList object to the user that contains all the possible data products that the user application may request based on the username and role. The user may then filter data products using the DataProductList object.

Parameters:

- subUser The user that is requesting this API call.

Returns:

- DDSXML_DataProductList The pointer to the data product filter list

2.2.1.3.17 DDSAPI_Message::getFilteredDataProductList

```
DDSXML_DataProductList
dds::RequestAPI::Message::DDSAPI_Message::getFilteredDataProductList (
    DDSXML_User subUser,
    DDSXML_RequestTypesEnum requestType
  )
```

This method returns a DDSXML_DataProductList object to the user that contain all the possible data products that the user application may request based on the username and role, restricted by those applicable to the supplied request type

Parameters:

- subUser The user that is requesting this API call.
- requestType The request type against which the products will initially be filtered

Returns:

- DDSXML_DataProductList The pointer to the data product filter list

2.2.1.3.18 DDSAPI_Message::createRequest

```
DDSXML_Request dds::RequestAPI::Message::DDSAPI_Message::createRequest (
    DDSXML_RequestTypesEnum requestType,
    DDSXML_ImplementationRequestTypesEnum implType,
    DDSXML_User user
```

)

This method instantiates a Request and returns a pointer to the instantiation.

Parameters:

- requestType The request type to use in this API call
- implType The implType type to use in this API call
- user The user that is requesting this API call.

Returns:

- DDSXML_Request A pointer to the newly constructed DDSXML_Request

2.2.1.3.19 DDSAPI_Message::validateRequestType

boolean dds::RequestAPI::Message::DDSAPI_Message::validateRequestType (

DDSXML_RequestTypesEnum requestType,

DDSXML_ImplementationRequestTypesEnum implType

)

Method to validate the request before submitting it. This will ensure that the request has the necessary parameters required for the request type input. Parameters may be found in Volume I of this document.

Parameters:

- requestType The request type to use in this API call
- implType The implType type to use in this API call

Returns:

- boolean True is this impl/type is valid for this user/role

2.2.1.3.20 DDSAPI_Message::createTemplateFromRequest

DDSXML_Request
dds::RequestAPI::Message::DDSAPI_Message::createTemplateFromRequest (

String requestID,

DDSXML_User subUser

)

This method creates a Template using data from an existing request and returns the template.

Parameters:

- requestID The ID of the template to be copied.
- subUser The user that is requesting this API call.

Returns:

- DDSXML_Request A pointer to the newly constructed DDSXML_Request

2.2.1.3.21 DDSAPI_Message::createRequestFromTemplate

DDSXML_Request
dds::RequestAPI::Message::DDSAPI_Message::createRequestFromTemplate (

String requestID,

DDSXML_User subUser

)

This method creates a Request using data from an existing template and returns the request.

Parameters:

- requestID The ID of the template to be copied.
- subUser The user that is requesting this API call.

Returns:

- DDSXML_Request The newly constructed DDSXML_Request

2.2.1.3.22 DDSAPI_Message::addRequest

boolean dds::RequestAPI::Message::DDSAPI_Message::addRequest (

DDSXML_Request request,

DDSXML_User subUser

)

This method adds the request to the Request Server. This can be any request, Template or modified request.

Parameters:

- request The request to be added to the Server.
- subUser The user that is requesting this API call.

Returns:

- bool - True if submitted OK, False if not

2.2.1.3.23 DDSAPI_Message::findRequest

DDSXML_Request dds::RequestAPI::Message::DDSAPI_Message::findRequest (

String requestID,

DDSXML_ImplementationRequestTypesEnum implType,
DDSXML_RequestTypesEnum requestType,

boolean templateFlag,

DDSXML_User subUser

)

This method returns the Request* that references the request that had a request ID equal to that of requestID.

Parameters:

- requestID The request ID for the request to be found.
- implType The implType type to use in this API call
- requestType The request type to use in this API call
- templateFlag True to verify if this is a template
- subUser The user that is requesting this API call.

Returns:

- DDSXML_Request a request found in the system that matches the ID passed in.

2.2.1.3.24 DDSAPI_Message::getRequests

Vector dds::RequestAPI::Message::DDSAPI_Message::getRequests (

DDSXML_ImplementationRequestTypesEnum implType,
DDSXML_RequestTypesEnum requestType,

boolean templateFlag,

boolean allRequests,

DDSXML_User subUser

)

This method will return all Requests for the user as a vector of Requests.

Parameters:

- implType The implType type to use in this API call
- requestType The request type to use in this API call
- templateFlag True to verify if this is a template
- allRequests True to add in all request
- subUser The user that is requesting this API call.

Returns:

- Vector The vector of user requests.

2.2.1.3.25 DDSAPI_Message::suspendRequest

boolean dds::RequestAPI::Message::DDSAPI_Message::suspendRequest (

String requestID,

long duration,

DDSXML_User subUser

)

This method suspends the request referenced by the requestID. The user may also provide a duration. If no values are given then all data products within the request are suspended indefinitely.

Parameters:

- requestID The ID of the request to be suspended.
- duration The duration for the suspension in microseconds. If this value is -1, the data product(s) will be suspended indefinitely.
- subUser The user that is requesting this API call.

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.26 DDSAPI_Message::resumeRequest

boolean dds::RequestAPI::Message::DDSAPI_Message::resumeRequest (

String requestID,

DDSXML_User subUser

)

This method resumes data products specified by the requestID.

Parameters:

- requestID The ID of the request to be resumed.
- subUser The user that is requesting this API call.

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.27 DDSAPI_Message::deleteRequest

boolean dds::RequestAPI::Message::DDSAPI_Message::deleteRequest (

String requestID,

DDSXML_User subUser

)

This method deletes a request from the system based on the request ID given as a parameter.

Parameters:

- requestID The ID of the request to be deleted.
- subUser The user that is requesting this API call.

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.28 DDSAPI_Message::deleteAllRequests

```
boolean dds::RequestAPI::Message::DDSAPI_Message::deleteAllRequests (
    DDSXML_User user
  )
```

This method deletes all requests from the system for the logged in user.

Parameters:

- user The user that is requesting this API call.

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.29 DDSAPI_Message::getNumberOfRequests

```
int dds::RequestAPI::Message::DDSAPI_Message::getNumberOfRequests (
    DDSXML_ImplementationRequestTypesEnum implType,
    DDSXML_RequestTypesEnum requestType,
    boolean templateFlag,
    DDSXML_User subUser
  )
```

This method returns the number of requests in the system.

Parameters:

- implType The implType type to use in this API call
- requestType The request type to use in this API call
- templateFlag True to verify if this is a template
- subUser The user that is requesting this API call.

Returns:

- int the number of requests in the system.

2.2.1.3.30 DDSAPI_Message::transferRequest

```
boolean dds::RequestAPI::Message::DDSAPI_Message::transferRequest (
    String requestID,
    DDSXML_User fromUser,
    DDSXML_User toUser
  )
```

This method transfers the request referenced by the requestID.

Parameters:

- requestID The ID of the request to be suspended.
- fromUser The user to transfer the request from
- toUser The user to transfer the request to

Returns:

- boolean True - if successful, False - if failed

2.2.1.3.31 DDSAPI_Message::isDQNEnabled

boolean dds::RequestAPI::Message::DDSAPI_Message::isDQNEnabled (

)

This method returns true if DQN is enabled.

Returns:

- boolean True – if DQN enabled

2.2.1.3.32 DDSAPI_Message::refresh

boolean dds::RequestAPI::Message::DDSAPI_Message::refresh (

)

This method will refresh the API from the Server. It may take some time after this returns before all of the data from the Server is in the API. This does not wait for all data before the return

Returns:

- boolean True – if refreshed OK

2.2.1.3.33 DDSAPI_Message::getUpdateIDs

Vector dds::RequestAPI::Message::DDSAPI_Message::getUpdateIDs (

)

This method returns all IDs of updated requests, catalogs and templates

Returns:

- Vector The vector containing the ID strings.

2.2.1.3.34 DDSAPI_Message::getDeletedIDs

Vector dds::RequestAPI::Message::DDSAPI_Message::getDeletedIDs (

)

This method returns all IDs of deleted requests, catalogs and templates

Returns:

- Vector The vector containing the ID strings.

2.2.1.3.35 DDSAPI_Message::getUsername

String dds::RequestAPI::Message::DDSAPI_Message::getUsername (

)

This method returns the username as a String.

Returns:

- String The username of the current user.

2.2.1.3.36 DDSAPI_Message::getVersion

String dds::RequestAPI::Message::DDSAPI_Message::getVersion (

)

This method returns the code/XML version as a String.

Returns:

- String The current version of the DDSAPI_Message

2.2.1.3.37 DDSAPI_Message::getClient

String dds::RequestAPI::Message::DDSAPI_Message::getClient (

)

This method returns the clientType as a String.

Returns:

- String The client type(i.e. API, GUI, HANDLER)

2.2.1.3.38 DDSAPI_Message::getRole

String dds::RequestAPI::Message::DDSAPI_Message::getRole (

)

This method returns the user's role as a String.

Returns:

- String The role that the user is currently logged in as.

2.2.1.3.39 DDSAPI_Message::getUser

DDSXML_User dds::RequestAPI::Message::DDSAPI_Message::getUser (

)

This method returns the user. Caller owns the user returned

Returns:

- DDSXML_User The DDS User structure.

2.2.1.3.40 DDSAPI_Message::getDomain

String dds::RequestAPI::Message::DDSAPI_Message::getDomain (

)

This method returns the domain of the Request Server.

Returns:

- String The Request Server's domain.

2.2.1.3.41 DDSAPI_Message::getSubDomain

String dds::RequestAPI::Message::DDSAPI_Message::getSubDomain (

)

This method returns the sub-domain of the Request Server.

Returns:

- String The Request Server's sub-domain.

2.2.1.3.42 DDSAPI_Message::getUsers

DDSXML_UserList dds::RequestAPI::Message::DDSAPI_Message::getUsers (

)

This method returns the sub users of this User(if any) .

Returns:

- DDSXML_UserList The Sub User list.

2.2.1.3.43 DDSAPI_Message::getTransferTypes

Vector dds::RequestAPI::Message::DDSAPI_Message::getTransferTypes (

)

This method returns all valid Transfer Types in a vector to the caller.

Returns:

- Vector The vector containing all valid Transfer Types for destinations.

2.2.1.3.44 DDSAPI_Message::validateTransferType

boolean dds::RequestAPI::Message::DDSAPI_Message::validateTransferType (

DDSXML_DestinationTransferTypeEnum transferType

)

This method validates transfer type.

Parameters:

- transferType The transferType for the destination FTP server.

Returns:

- boolean True if OK, False if not

2.2.1.3.45 Deleted

2.2.1.3.46 DDSAPI_Message::addDestinationSetEntry

String dds::RequestAPI::Message::DDSAPI_Message::addDestinationSetEntry (

destinationName,

hostName,

path,

ftpUserName,

ftpPassword,

transferType,

_User subUser

)

This method adds a new destination to the user defined destination list.

Parameters:

- String destinationName,
- String hostName,
- String path,
- String ftpUserName,
- String ftpPassword,
- DDSXML_DestinationTransferTypeEnum transferType,
- DDSXML_User subUser

Returns:

- String Destination ID or "" if there was an error

2.2.1.3.47 DDSAPI_Message::deleteDestinationSetEntry

boolean dds::RequestAPI::Message::DDSAPI_Message::deleteDestinationSetEntry (

destination

destinationEntryID

subUser

)

This method removes a user defined destination set from the user destination and if the last entry then removes the destination from the destination list.

Parameters:

- DDSXML_Destination destination,
- String destinationEntryID,
- DDSXML_User subUser.

Returns:

- boolean True - if successful, False - if failed.

2.2.1.3.48 DDSAPI_Message::modifyDestinationSetEntry

boolean dds::RequestAPI::Message::DDSAPI_Message:: modifyDestinationSetEntry (

destination

destinationEntry

subUser

)

This method modifies the Destination Entry in the Destination.

Parameters:

- DDSXML_Destination destination,
- DDSXML_DestinationEntry destinationEntry,

- DDSXML_User subUser

Returns:

- boolean True - if successful, False - if failed.

2.2.1.3.49 DDSAPI_Message::moveDestinationSetEntry

boolean dds::RequestAPI::Message::DDSAPI_Message::moveDestinationSetEntry (

originalDestination

originalEntryID

newDestination

newEntry

subUser

)

This method moves the Destination Entry from one destination to another destination.

Parameters:

- DDSXML_Destination originalDestination,
- String originalEntryID,
- DDSXML_Destination newDestination,
- DDSXML_DestinationEntry newEntry,
- DDSXML_User subUser

Returns:

- boolean True - if successful, False - if failed.

2.2.1.3.50 DDSAPI_Message::getLoginConfigurationState

DDSAPI_LoginStateEnum
dds::RequestAPI::Message::DDSAPI_Message::getLoginConfigurationState (

)

This method returns the login configuration state of the system. If ALL_CONFIG_STATE_SUCCESSFUL_ENUM then the API has been fully configured. If not then a part or all of the API has not been configured. The enum returned is the current state of API configuration The API must be fully configured to use the other methods in the API.

Returns:

- DDSAPI_LoginStateEnum The login configuration state, ALL_CONFIG_STATE_SUCCESSFUL_ENUM - configured - else not configured

2.2.1.3.51 DDSAPI_Message::getDataShipmentList

DDSXML_DataShipmentList
dds::RequestAPI::Message::DDSAPI_Message::getDataShipmentList (

String requestID,

int startingRecordOffset,

int maxNumberOfRecords,

DDSXML_User user

)

This method returns a DDSXML_DataShipmentList to the user.

A empty list or a list less than max Number Of Records will be returned if there are no more DSR records left. Multiple calls to this can be made incrementing the starting record offset by the size of the list returned each time. (Not by maxNumberOfRecords which may be limited internally) You can then add the multiple lists together.

Parameters:

- requestID The ID of the DDSXML_Request to get the shipment records for.
- startingRecordOffset The starting record offset
- maxNumberOfRecords The max number of records to transfer This is limited inside the API/Server if too large
- user The user that is requesting this API call.

Returns:

- DDSXML_DataShipmentList The list object received by the API. Returns null on error

2.2.1.3.52 DDSAPI_Message::getRequestDestinations

Vector<DDSXML_Destination>
dds::RequestAPI::Message::DDSAPI_Message::getRequestDestinations (

String requestID,

DDSXML_User subUser

)

This method returns all the user destinations for a specific request belonging to a subuser of the current supervisor. This will also work on a request belonging to the current user, but getDestinations will retrieve those destinations faster.

Parameters:

- requestID The request for which to retrieve destinations
- subUser The user that is requesting this API call.

Returns:

- Vector<DDSXML_Destination> The vector containing the user destinations.

2.2.1.3.53 DDSAPI_Message::getSubuserDestinations

DDSXML_DestinationList
dds::RequestAPI::Message::DDSAPI_Message::getSubuserDestinations (

)

This method returns all the subusers defined destinations in an XML object to the caller.

Returns:

- DDSXML_DestinationList The object containing all user-defined destinations.

2.2.1.3.54 DDSAPI_Message::deleteSubuserDestination

boolean dds::RequestAPI::Message::DDSAPI_Message::deleteSubuserDestination (

DDSXML_Destination destination

)

This method removes a subuser's defined destination from that user's destination list.

Parameters:

- destination The destination to be removed from the destination list.

Returns:

boolean True - if successful, False - if failed

2.2.2 DDSXML_CatalogRequest Class Reference

This is the XML data class for the Catalog. This class is responsible for storing and maintaining the state of a request in the system.

The Class diagram representing the DDSXML_CatalogRequest Class is provided in Figure 2.2.2-1, DDSXML_CatalogRequest Class UML Diagram.

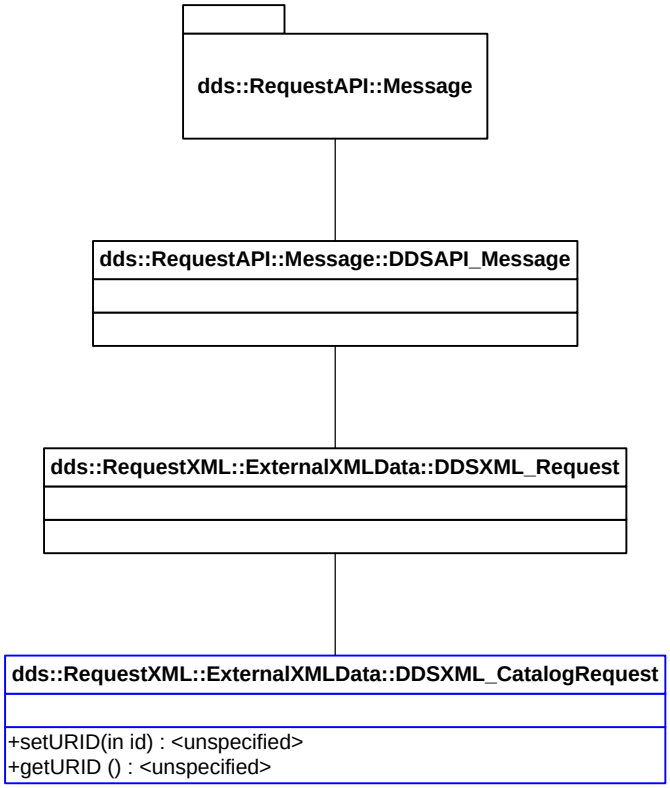

Figure 2.2.2-1, DDSXML_CatalogRequest Class UML Diagram

2.2.2.1 DDSXML_CatalogRequest Class Functions

2.2.2.1.1 DDSXML_CatalogRequest::setURID

boolean dds::RequestXML::ExternalXMLData::DDSXML_CatalogRequest::setURID (

String id

)

Set the catalog URID to query on.

Parameters:

- id The catalog URID to query on.

Returns:

- boolean True if OK, else false

2.2.2.1.2 DDSXML_CatalogRequest::getURID

String dds::RequestXML::ExternalXMLData::DDSXML_CatalogRequest::getURID (

)

Obtains the catalog's URID and returns it.

Returns:

- String The catalog's URID as a string Empty string is not found.

2.2.3 Deleted

2.2.4 DDSXML_DataProductList Class Reference

This class is responsible for handling the data product List XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. This is used for the initial configuration from the Server. The request uses an ID list only.

The Class diagram representing the DDSXML_DataProductList Class is provided in Figure 2.2.4-1, DDSXML_DataProductList Class UML Diagram.

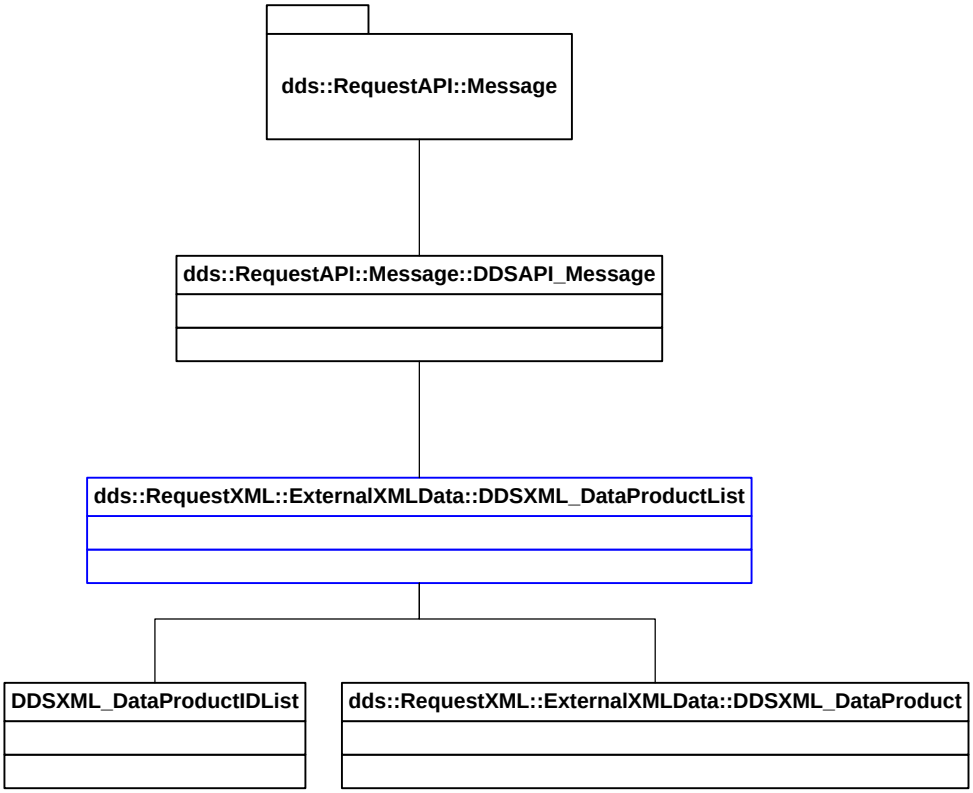

Figure 2.2.4-1, DDSXML_DataProductList Class UML Diagram

2.2.4.1 DDSXML_DataProductList Class Functions

2.2.4.1.1 DDSXML_DataProductList::addDataProduct

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::addDataProduct (

DDSXML_DataProduct dataProduct

)

This method adds the data Product to this list.

Parameters:

- dataProduct The data Product

Returns:

- boolean True if OK, else false

2.2.4.1.2 DDSXML_DataProductList::deleteDataProduct

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::deleteDataProduct (

String dataProductID

)

Allows the caller to delete a data product from the list.

Parameters:

- dataProductID Deletes data product based on data product ID.

Returns:

- boolean true if the deletion was successful.

2.2.4.1.3 DDSXML_DataProductList::getDataProductsSet

dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getDataProductsSet
(

)

This method returns the DataProducts.

Returns:

- Set<DataProduct > The Data Product's

2.2.4.1.4 DDSXML_DataProductList::getDataProducts

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getDataProducts (

)

This method returns the DataProducts.

Returns:

- Vector A vector containing the Data Products

2.2.4.1.5 DDSXML_DataProductList::getDataProductList

DDSXML_DataProductList
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getDataProductList
(

DDSXML_DataProductIDList dataProductIDList

)

This method returns the DataProducts for the DataProduct ID's passed in.

Parameters:

- dataProductIDList The list of Data product ID's to get Data products for.

Returns:

- DDSXML_DataProductList The Data Product's

2.2.4.1.6 DDSXML_DataProductList::getFilteredList

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getFilteredList (

String shortname,

String spacecraft,

String sensor,

String requestType,

String productType

)

This method returns a filtered Vector of DataProduct pointers that match the parameters provided by the user. If no data product(s) match the parameters provided, an empty Vector is returned.

Parameters:

- shortname The Shortname of the Data Product. The empty String represents all shortnames.
- spacecraft The spacecraft. The empty String represents all spacecraft.
- sensor The sensor. The empty String represents all sensors.
- productType The product type. The empty String represents all types.
- requestType The request type. The empty String represents all categories.

Returns:

- Vector The Vector of DDSXML_DataProducts that contains all the Data Products, requestable by the user, that meet the filter criteria.

2.2.4.1.7 DDSXML_DataProductList::getFilteredDataProductList

DDSXML_DataProductList
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getFilteredDataProductList (

String shortname,

String spacecraft,

String sensor,

String requestType,

String productType

)

This method returns a filtered Vector of DataProduct pointers that match the parameters provided by the user. If no data product(s) match the parameters provided, an empty Vector is returned.

Parameters:

- shortname The Shortname of the Data Product. The empty String represents all shortnames.
- spacecraft The spacecraft. The empty String represents all spacecraft.
- sensor The sensor. The empty String represents all sensors.
- requestType The request type. The empty String represents all categories.
- productType The product type. The empty String represents all types.

Returns:

- Vector The Vector of DDSXML_DataProducts that contains all the Data Products, requestable by the user, that meet the filter criteria.

2.2.4.1.8 DDSXML_DataProductList::getFilteredList

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getFilteredList (

String shortname,

String spacecraft,

String sensor,

String productType

)

This method returns a filtered Vector of DataProduct pointers that match the parameters provided by the user. If no data product(s) match the parameters provided, an empty Vector is returned.

Parameters:

- shortname The Shortname of the Data Product. The empty String represents all shortnames.
- spacecraft The spacecraft. The empty String represents all spacecraft.
- sensor The sensor. The empty String represents all sensors.
- productType The product type. The empty String represents all types.

Returns:

- Vector The Vector of DDSXML_DataProducts that contains all the Data Products, requestable by the user, that meet the filter criteria.

2.2.4.1.9 DDSXML_DataProductList::getShortnames

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getShortnames (

)

This method returns a Vector of Shortnames that can be used to query as a parameter in getFilteredList.

Returns:

- Vector A Vector of Strings that contain all possible, requestable data product shortnames.

2.2.4.1.10 DDSXML_DataProductList::getSpacecrafts

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getSpacecrafts (

)

This method returns a Vector of Spacecraft that can be used to query as a parameter in getFilteredList.

Returns:

- Vector A Vector of Strings that contain all possible, requestable spacecraft.

2.2.4.1.11 DDSXML_DataProductList::getSpacecraftsSet

Set
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getSpacecraftsSet (

Vector productList

)

This method returns a Vector of Spacecraft that can be used to query as a parameter in getFilteredList.

Parameters:

- productList The product list

Returns:

- Set A Set of Strings that contain all possible, requestable spacecraft.

2.2.4.1.12 DDSXML_DataProductList::getSensors

Vector dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getSensors (

)

This method returns a Vector of Sensors that can be used to query as a parameter in getFilteredList.

Returns:

- Vector A Vector of Strings that contain all possible, requestable categories.

2.2.4.1.13 DDSXML_DataProductList::getSensorsSet

Set dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getSensorsSet (

Vector productList

)

This method returns a Vector of Sensors that can be used to query as a parameter in getFilteredList.

Parameters:

- productList The product list

Returns:

- Set A Set of Strings that contain all possible, requestable Sensors.

2.2.4.1.14 DDSXML_DataProductList::getRequestTypes

Vector dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getRequestTypes (

)

This method returns a Vector of distinct request types in the product list

Returns:

- Vector of Strings representing the distinct request types in product list

Deprecated:

Request types are pre-filtered now

2.2.4.1.15 DDSXML_DataProductList::getProductTypes

Vector dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getProductTypes (

)

This method returns a Vector of distinct product types in the product list

Returns:

- Vector of Strings representing the distinct product types in product list

2.2.4.1.16 DDSXML_DataProductList::getProductTypesSet

Set dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getProductTypesSet (

Vector productList

)

This method returns a Vector of Product Types that can be used to query as a parameter in getFilteredList.

Parameters:

- productList The product list

Returns:

- Set A Set of Strings that contain all possible, requestable Product Types.

2.2.4.1.17 DDSXML_DataProductList::getDataProduct

DDSXML_DataProduct
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getDataProduct (

String dataProductID

)

This method returns the dataProduct identified by the dataProduct ID if it exists in the system

Parameters:

- dataProductID The dataProductID to check for

Returns:

- DDSXML_DataProduct The dataProduct.

2.2.4.1.18 DDSXML_DataProductList::hasDataProductID

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::hasDataProductID (

String dataProductID

)

This method returns true if the dataProduct ID exists in the system

Parameters:

- dataProductID The dataProductID to check for

Returns:

- boolean True if the data product ID exists, false otherwise.

2.2.4.1.19 DDSXML_DataProductList::getNumberOfDataProducts

int
dds::RequestXML::ExternalXMLData::DDSXML_DataProductList::getNumberOfDataProducts (

)

This method returns the number of dataProducts for this user

Returns:

- int The number of dataProducts for this user

2.2.5 DDSXML_DataShipment Class Reference

This class is responsible for handling the Data Product Message XML.

It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. Some of the method names are left to be compatible with the old API. This class was(is) exposed to external users so we can not change the method names now.

The Class diagram representing the DDSXML_DataShipment Class is provided in Figure 2.2.5.-1, DDSXML_DataShipment Class UML Diagram.

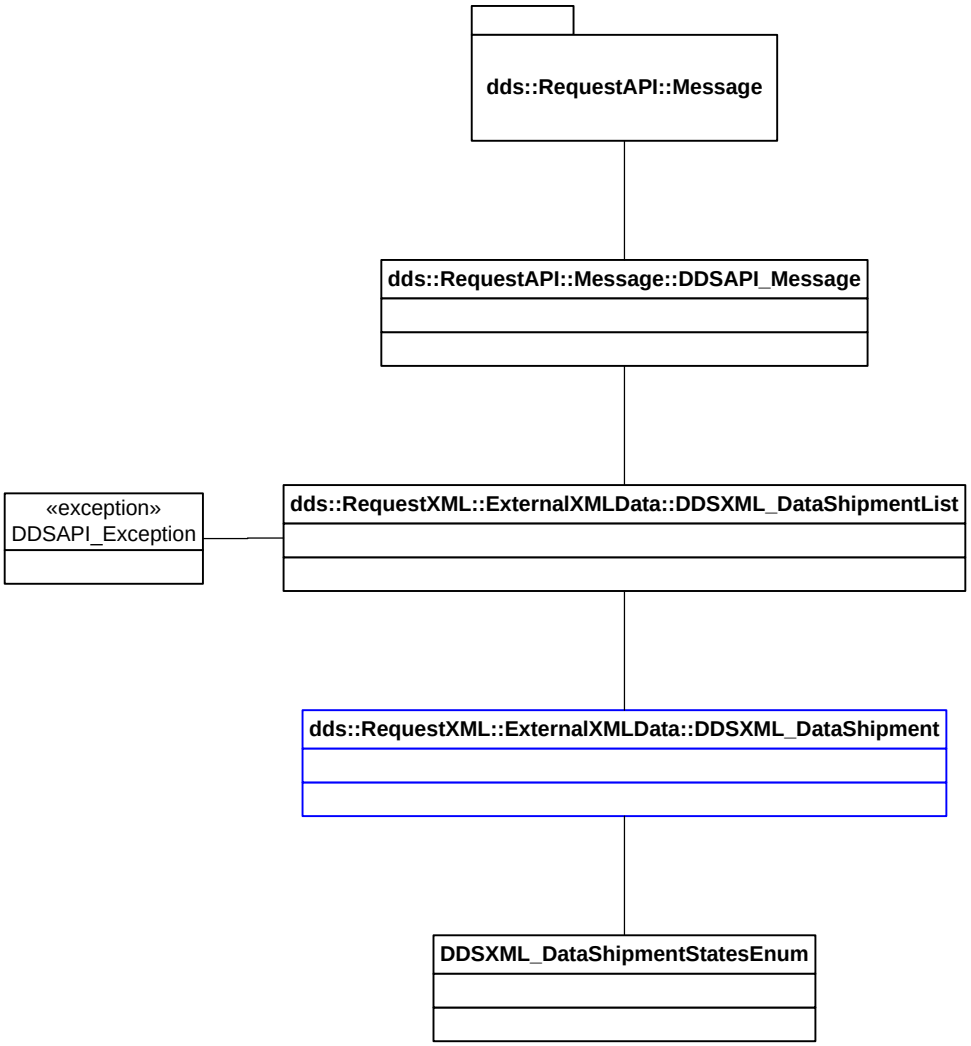

Figure 2.2.5-1, DDSXML_DataShipment Class UML Diagram

2.2.5.1 DDSXML_DataShipment Class Functions

2.2.5.1.1 DDSXML_DataShipment::getRequestID

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getRequestID ( )

This method retrieves the Request ID as a string.

Returns:

- std::string The shipment Name or "" if not valid.

2.2.5.1.2 DDSXML_DataShipment::getMessage

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getMessage ( )

This method retrieves the message as a string.

Returns:

- String The message

2.2.5.1.3 DDSXML_DataShipment::getURID

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getURID ( )

This method retrieves the myURID attribute's value as a string.

Returns:

- std::string The shipment path as a string or "" if not valid.

2.2.5.1.4 DDSXML_DataShipment::getFileName

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getFileName ( )

This method retrieves the FileName attribute's value as a string.

Returns:

- std::string The FileName associated with this shipment or "" if not valid.

2.2.5.1.5 DDSXML_DataShipment::getTimeStamp

long dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getTimeStamp ( )

This method retrieves the myTimeStamp attribute's value as a string.

Returns:

- std::string The timestamp associated with this shipment or "" if not valid.

2.2.5.1.6 DDSXML_DataShipment::getTransferTime

long dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getTransferTime (

)

This method retrieves the transfer time of this shipment

Returns:

- long The transfer time of this shipment.

2.2.5.1.7 DDSXML_DataShipment::getDataShipmentState

DDSXML_DataShipmentStatesEnum dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getDataShipmentState (

)

This method retrieves the last state of this shipment This will allow the shipment to be validated in the future

Returns:

- DDSXML_DataShipmentStatesEnum The state associated with this shipment

2.2.5.1.8 DDSXML_DataShipment::getFTPUserName

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getFTPUserName (

)

This method retrieves the FTPUserName attribute's value as a string.

Returns:

- String The FTPUserName associated with this shipment or "" if not valid.

2.2.5.1.9 DDSXML_DataShipment::getCollectionShortName

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getCollectionShortName (

)

This method retrieves the Collection Short Name attribute's value as a string.

Returns:

- String The Collection Short Name associated with this shipment or "" if not valid.

2.2.5.1.10 Deleted

2.2.5.1.11 DDSXML_DataShipment::getHostName

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getHostName (

)

This method retrieves the HostName attribute's value as a string.

Returns:

- String The HostName associated with this shipment or "" if not valid.

2.2.5.1.12 DDSXML_DataShipment::getFilePath

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getFilePath (

)

This method retrieves the FilePath attribute's value as a string.

Returns:

- String The FilePath associated with this shipment or "" if not valid.

2.2.5.1.13 DDSXML_DataShipment::getCheckSum

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getCheckSum (

)

This method retrieves the CheckSum attribute's value as a string.

Returns:

- String The CheckSum associated with this shipment or "" if not valid.

2.2.5.1.14 DDSXML_DataShipment::getFileSize

int dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getFileSize (

)

This method retrieves the FileSize attribute's value as a string.

Returns:

- String The FileSize associated with this shipment or "" if not valid.

2.2.5.1.15 DDSXML_DataShipment::getDestinationID

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getDestinationID (

)

This method retrieves the Destination ID attribute's value as a string.

Returns:

- String The Destination ID associated with this shipment or "" if not valid.

2.2.5.1.16 DDSXML_DataShipment::getDestinationName

String dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getDestinationName (

)

This method retrieves the Destination Name attribute's value as a string.

Returns:

- String The Destination Name associated with this shipment or "" if not valid.

2.2.5.1.17 DDSXML_DataShipment::getMasterDestinationIndex

String
dds::RequestXML::ExternalXMLData::DDSXML_DataShipment::getMasterDestinationIndex (

)

This method retrieves the Master Destination Index attribute's value as a string.

Returns:

- String The Master Destination Index associated with this shipment or "" if not valid.

2.2.6 DDSXML_DataShipmentList Class Reference

This class is responsible for handling the data shipment list XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API.

The Class diagram representing the DDSXML_DataShipmentList Class is provided in Figure 2.2.6.-1, DDSXML_DataShipmentList Class UML Diagram.

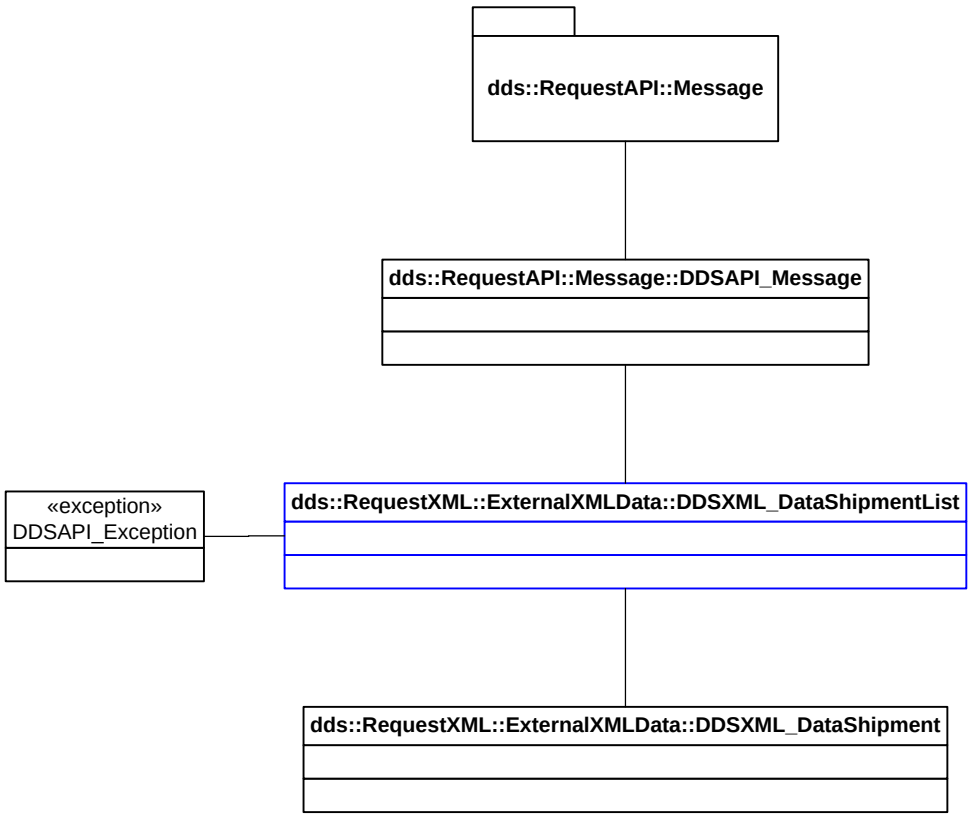

Figure 2.2.6-1, DDSXML_DataShipmentList Class UML Diagram

2.2.6.1 DDSXML_DataShipmentList Class Functions

2.2.6.1.1 DDSXML_DataShipmentList::addDataShipment

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::addDataShipment
(

DDSXML_DataShipment dataShipment

) throws DDSAPI_Exception

Add a user DATA_SHIPMENT to the shipment list.

Parameters:

- dataShipment The new shipment to add.

Returns:

- boolean True if OK, false if not.

2.2.6.1.2 DDSXML_DataShipmentList::deleteDataShipment

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::deleteDataShipment (

String key

)

Allows the caller to delete a user shipment from the list.

Parameters:

- key Deletes the user shipment based upon the key.

Returns:

- bool True if OK, false if not.

2.2.6.1.3 DDSXML_DataShipmentList::deleteLastRecord

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::deleteLastRecord
(

int numberToDelete

)

Allows the caller to delete a user shipment from the list.

Parameters:

- numberToDelete The number of records to delete from the end.

Returns:

- boolean True if OK, false if not.

2.2.6.1.4 DDSXML_DataShipmentList::deleteAllRecords

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::deleteAllRecords
(

)

Allows the caller to delete a user shipment from the list.

Returns:

- boolean True if OK, false if not.

2.2.6.1.5 DDSXML_DataShipmentList::getDataShipments

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::getDataShipments
(

)

Get the list of all user DataShipments as a vector.

Returns:

- Vector The vector that contains all user shipments. The DDSXML_DataShipmentSystem objects referenced by this vector are owned by the caller and should be destroyed. This vector may become invalid if a shipment is deleted from the list.

2.2.6.1.6 DDSXML_DataShipmentList::getDataShipment

DDSXML_DataShipment
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::getDataShipment
(

String key

)

This method returns the record identified by the shipment ID if it exists in the system

Parameters:

- key The index to check for

Returns:

- DDSXML_DataShipment The data shipment.

2.2.6.1.7 DDSXML_DataShipmentList::hasDataShipmentID

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::hasDataShipmentID (

String index

)

This method returns true if the shipment ID exists in the system

Parameters:

- index The index to check for

Returns:

- boolean True if found, else false

2.2.6.1.8 DDSXML_DataShipmentList::getNumberOfDataShipments

int dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::getNumberOfDataShipments (

)

This method returns the number of shipments in this list

Returns:

- int The number of shipments in this list

2.2.6.1.9 DDSXML_DataShipmentList::limitDataShipments

boolean dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::limitDataShipments (

int limit

)

This method limits the number of shipments for this user

Parameters:

- limit The number of shipments for this user

Returns:

- boolean True if limit is set

Deprecated:

Use setDSRLimit

2.2.6.1.10 DDSXML_DataShipmentList::getDSRLimit

int dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::getDSRLimit (

)

This method returns the maximum DSRs per list. This limit may have been lowered by the server

Returns:

- int The DSRLimit

2.2.6.1.11 DDSXML_DataShipmentList::getStartingOffsetOfDSRList

int
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::getStartingOffset
OfDSRList (

)

This method returns the offset of the first DSR in this list

Returns:

- int The StartingOffsetOfDSRList

2.2.6.1.12 DDSXML_DataShipmentList::getMaxNumberOfDSRAvailable

int
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::getMaxNumberOf
DSRAvailable (

)

This method returns the number of DSRs that currently exist for this request

Returns:

- int The MaxNumberOfDSRAvailable

2.2.6.1.13 DDSXML_DataShipmentList::setDSRLimit

void dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::setDSRLimit
(

int DSRLimit

)

This method sets the DSRLimit

Parameters:

- DSRLimit The DSRLimit

2.2.6.1.14 DDSXML_ DataShipmentList::setStartingOffsetOfDSRList

void
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::setStartingOffset
OfDSRList (

int StartingOffsetOfDSRList

)

This method sets the StartingOffsetOfDSRList

Parameters:

- int StartingOffsetOfDSRList

2.2.6.1.15 DDSXML_ DataShipmentList::setMaxNumberOfDSRAvailable

void
dds::RequestXML::ExternalXMLData::DDSXML_DataShipmentList::setMaxNumberOfDSRAvailable (

int MaxNumberOfDSRAvailable

)

This method sets the MaxNumberOfDSRAvailable

Parameters:

- MaxNumberOfDSRAvailable The MaxNumberOfDSRAvailable

2.2.7 DDSXML_Destination Class Reference

This class is responsible for handling the user destination XML.

This represents both a USER destination and a MASTER destination at the same time. This contains a Master index ID for the master destination. The can be multiple user destinations pointing to the same master destination ID. Each Master destination has a set of one or more destination entries. Master destinations are for internal use and must not be changed outside of the system.

This allows the user destination to be transfered/copied to another user and also allows for an destination entry change to be reflected in all the different user destinations that point to the same master destination.

Entry method names are left to be compatible with the old API. So accessing this destination for the entry items always points to the first destination entry. All destinations have at least one destination entry.

Each destination entry can also be accessed by using getDestinationEntry to get the Entry and then the attribute needed. Destination Entries can not be modified outside of the API. A modify destination (or modify destination set) command must be issued to modify destination set

It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. Some of the method names are left to be compatible with the old API. This class was(is) exposed to external users so we can not change the method names now.

The Class diagram representing the DDSXML_Destination Class is provided in Figure 2.2.7.-1, DDSXML_Destination Class UML Diagram.

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

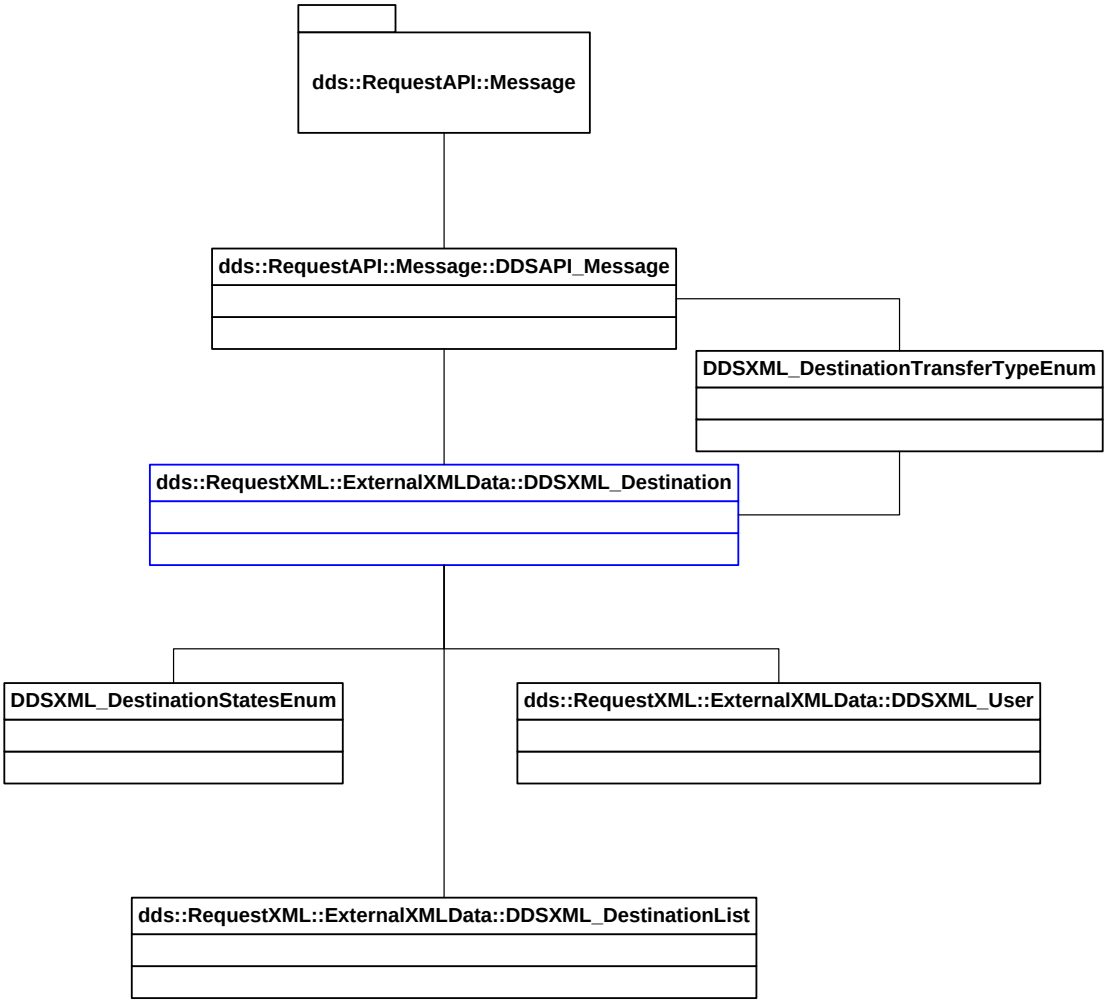

Figure 2.2.7-1, DDSXML_Destination Class UML Diagram

2.2.7.1 DDSXML_Destination Class Functions

2.2.7.1.1 DDSXML_Destination::getDestinationName

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getDestinationName (

)

This method retrieves the Destination Name as a string.

Returns:

- String The destination Name or "" if not valid.

2.2.7.1.2 DDSXML_Destination::getUserPassword

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getUserPassword (

)

This method retrieves the myPassword attribute's value as a string. The value returned will be clear text.

Returns:

- String The password as a string or "" if not valid.

2.2.7.1.3 DDSXML_Destination::getPath

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getPath (

)

This method retrieves the myPath attribute's value as a string.

Returns:

- String The destination path as a string or "" if not valid.

2.2.7.1.4 DDSXML_Destination::getHostName

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getHostName (

)

This method returns the destination host name (or IP) Note that this is really a valid hostname or IP address that can be used for FTP.

Returns:

- String The Host name as a string or "" if not valid.

2.2.7.1.5 DDSXML_Destination::getUserName

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getUserName (

)

This method retrieves the myUsername attribute's value as a string.

Returns:

- String The username associated with this destination or "" if not valid.

2.2.7.1.6 DDSXML_Destination::getOwner

DDSXML_User
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getOwner (

)

This method retrieves the owner of this destination

Returns:

- DDSXML_User The owner of this destination or NULL if not valid.

2.2.7.1.7 DDSXML_Destination::getState

DDSXML_DestinationStatesEnum
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getState (

)

This method retrieves the last state of this destination

Returns:

- DDSXML_DestinationStatesEnum The state associated with this destination

2.2.7.1.8 DDSXML_Destination::getTransferType

DDSXML_DestinationTransferTypeEnum
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getTransferType (

)

This method retrieves the transfer Type of this destination This will allow the destination to be validated in the future

Returns:

- DDSXML_DestinationTransferTypeEnum The transfer Type associated with this destination

2.2.7.1.9 DDSXML_Destination::setDestinationName

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Destination::setDestinationName (

String destinationName

)

This method sets the name of the destination. This is a user defined name.

Parameters:

- destinationName The name used to set destination name.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.10 DDSXML_Destination::setUserPassword

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Destination::setUserPassword (

String password

)

This method sets the myPassword attribute in the Destination class.

Parameters:

- password The password being used to set myPassword.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.11 DDSXML_Destination::setPath

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setPath (

String path

)

This method sets the myPath attribute in the Destination class.

Parameters:

- path The path being used to set myPath.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.12 DDSXML_Destination::setHostName

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setHostName (

String hostName

)

This method sets the destination host name Note that this is really a valid hostname or IP address that can be used for FTP.

Parameters:

- hostName The host name.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.13 DDSXML_Destination::setUserName

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setUserName (

String userName

)

This method sets the FTP user.

Parameters:

- userName The user name.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.14 DDSXML_Destination::setOwner

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setOwner (

DDSXML_User owner

)

This method sets The owner of this destination

Parameters:

- owner The owner of this destination.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.15 DDSXML_Destination::setState

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setState (

DDSXML_DestinationStatesEnum state

)

This method sets the state attribute in the Destination class. This will allow the destination to be validated in the future

Parameters:

- state The state of the Destination.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.16 DDSXML_Destination::setTransferType

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setTransferType (

DDSXML_DestinationTransferTypeEnum transferType

)

This method sets the transfer Type attribute in the Destination class. This will allow the destination to be sent using different transfer methods.

Parameters:

- transferType The transfer Type of the Destination.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.17 DDSXML_Destination::getUserIndex

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getUserIndex (

)

This method returns the user Destination index

Returns:

- String The user Index as a string or "" if not valid.

2.2.7.1.18 DDSXML_Destination::getMasterIndex

String dds::RequestXML::ExternalXMLData::DDSXML_Destination::getMasterIndex (

)

This method returns the master Destination index

Returns:

- String The Master Index as a string or "" if not valid.

2.2.7.1.19 DDSXML_Destination::setFTPUser

boolean dds::RequestXML::ExternalXMLData::DDSXML_Destination::setFTPUser (

String userName,

String userPassword

)

This method sets the FTP user.

Parameters:

- userName The user name used to set myUsername.
- userPassword The password being used to set myPassword.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.20 DDSXML_Destination::getAllStates

DDSXML_DestinationStatesEnum
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getAllStates (

)

This method retrieves the state of all the destination entries

Returns:

- DDSXML_DestinationStatesEnum Valid only if all entries are valid

2.2.7.1.21 DDSXML_Destination::getSubsetCount

long dds::RequestXML::ExternalXMLData::DDSXML_Destination::getSubsetCount (

)

Get the number of entries in this destination.

Returns:

- long the mySubsetCount

2.2.7.1.22 DDSXML_Destination::getDestinationEntries

Vector<DDSXML_DestinationEntry>
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getDestinationEntries (

boolean copyflag

)

Get the list of all destination entries as a vector.

Parameters:

- copyflag true to get a copy of the entries

Returns:

- Vector<DDSXML_DestinationEntry> The vector that contains all the user destinations. This vector may become invalid if a destination is deleted from the list.

2.2.7.1.23 DDSXML_Destination::getDestinationEntry

DDSXML_DestinationEntry
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getDestinationEntry (

String entryID,

boolean copyDestination

)

This method returns the destination identified by the entry ID if it exists in the system If the ID is 0 then the first entry is returned

Parameters:

- entryID The entryID to check for
- copyDestination Whether to return a copy of the destination entry or the original entry object.

Returns:

- DDSXML_DestinationEntry A pointer to the destination . 0 - The destination was not found. Valid pointer otherwise.

2.2.7.1.24 DDSXML_Destination::getFirstDestinationEntry

DDSXML_DestinationEntry
dds::RequestXML::ExternalXMLData::DDSXML_Destination::getFirstDestinationEntry (

boolean copyDestination

)

This method returns the first destination entry for this destination

Parameters:

- copyDestination Whether to return a copy of the destination entry or the original entry object.

Returns:

- DDSXML_DestinationEntry A pointer to the destination . 0 - The destination was not found. Valid pointer otherwise.

2.2.7.1.25 DDSXML_Destination::addDestinationEntry

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Destination::addDestinationEntry (

DDSXML_DestinationEntry destinationEntry

)

Add an entry to this destination.

Parameters:

- destinationEntry The new destination entry to add.

Returns:

- boolean True if set OK, False if not.

2.2.7.1.26 DDSXML_Destination::deleteDestinationEntry

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Destination::deleteDestinationEntry (

String destinationEntryID

)

Allows the caller to delete a destination entry from the destination.

Parameters:

- destinationEntryID Deletes the destination entry based upon the entry id.

Returns:

- boolean True if OK, False if not.

2.2.7.1.27 DDSXML_Destination::modifyDestinationEntry

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Destination::modifyDestinationEntry (

DDSXML_DestinationEntry destinationEntry

)

Modify or add a single destination entry.

Parameters:

- destinationEntry The modified/new destination entry.

Returns:

- boolean True if OK, False if not.

2.2.7.1.28 DDSXML_Destination::hasDestinationEntryID

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Destination::hasDestinationEntryID (

String destinationEntryID

)

This method returns true if the destination Entry ID exists in the system

Parameters:

- destinationEntryID The destinationEntryID to check for

Returns:

- boolean True if the destination entry is found.

2.2.8 DDSXML_DestinationList Class Reference

This class is responsible for handling the Destination List XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API.

The Class diagram representing the DDSXML_Destination Class is provided in Figure 2.2.8.-1, DDSXML_DestinationList UML Diagram.

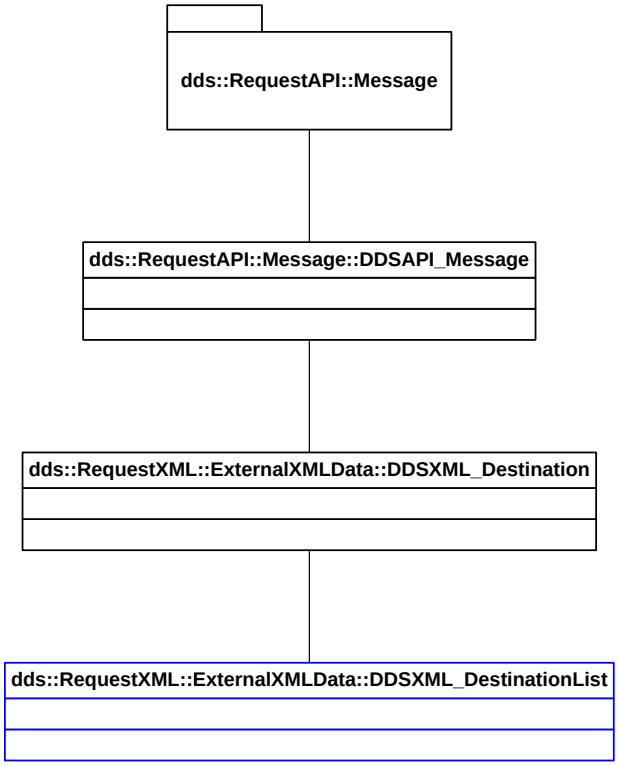

Figure 2.2.8-1, DDSXML_DestinationList Class UML Diagram

2.2.8.1 DDSXML_DestinationList Class Functions

2.2.8.1.1 DDSXML_DestinationList::addDestination

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::addDestination (

DDSXML_Destination destination

)

Add a user destination to the destination list.

Parameters:

- destination The new destination to add.

Returns:

- boolean True if OK, false if not.

2.2.8.1.2 DDSXML_DestinationList::deleteDestination

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::deleteDestination (

String destinationID

)

Allows the caller to delete a user destination from the list.

Parameters:

- destinationID Deletes the user destination based upon the destination id.

Returns:

- boolean True if OK, false if not.

2.2.8.1.3 DDSXML_DestinationList::getDestinations

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::getDestinations (

)

Get the list of all user Destinations as a vector.

Returns:

- Vector The vector that contains all user destinations. This vector may become invalid if a destination is deleted from the list.

2.2.8.1.4 DDSXML_DestinationList::modifyDestination

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::modifyDestination (

DDSXML_Destination destination

)

Modify the contents of a single user destination.

Parameters:

- destination The modified/new destination.

Returns:

- boolean True if OK, false if not.

2.2.8.1.5 DDSXML_DestinationList::getDestination

.DDSXML_Destination
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::getDestination (

String destinationID

)

This method returns the destination identified by the destination ID if it exists in the system

Parameters:

- destinationID The destinationID to check for

Returns:

- DDSXML_Destination The user destination.

2.2.8.1.6 DDSXML_DestinationList::hasDestinationID

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::hasDestinationID (

String destinationID

)

This method returns true if the destination ID exists in the system

Parameters:

- destinationID The destinationID to check for

Returns:

- boolean True if the destination is found.

2.2.8.1.7 DDSXML_DestinationList::hasDestination

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::hasDestination (

DDSXML_Destination destination

boolean checkId

)

This method returns true if the destination exists in the list

Parameters:

- destination The destination to check for
- checkID If true check the ID

Returns:

- boolean true if found.

2.2.8.1.8 DDSXML_DestinationList::getNumberOfDestinations

int
dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::getNumberOfDestinations (

)

This method returns the number of destinations for this user

Returns:

- int The number of destinations for this user

2.2.8.1.9 Deleted

2.2.8.1.10 Deleted

2.2.8.1.11 DDSXML_DestinationList:: getRequestDestinations

Vector dds::RequestXML::ExternalXMLData::DDSXML_DestinationList::
getRequestDestinations (

int minimumDestinationSets

)

Get the list of all user Request Destinations as a vector.

Parameters:

- minimumDestinationSets The minimum destinations in a destination set

Returns:

- Vector The vector that contains all user Request destinations. This vector may become invalid if a destination is deleted from the list.

Deprecated:

Identical to getDestinations

2.2.9 Deleted

2.2.10 DDSXML_GEORequest Class Reference

This is the XML data class for the Catalog. This class is responsible for storing and maintaining the state of a request in the system.

The Class diagram representing the DDSXML_GEORequest Class is provided in Figure 2.2.10.-1, DDSXML_GEORequest UML Diagram.

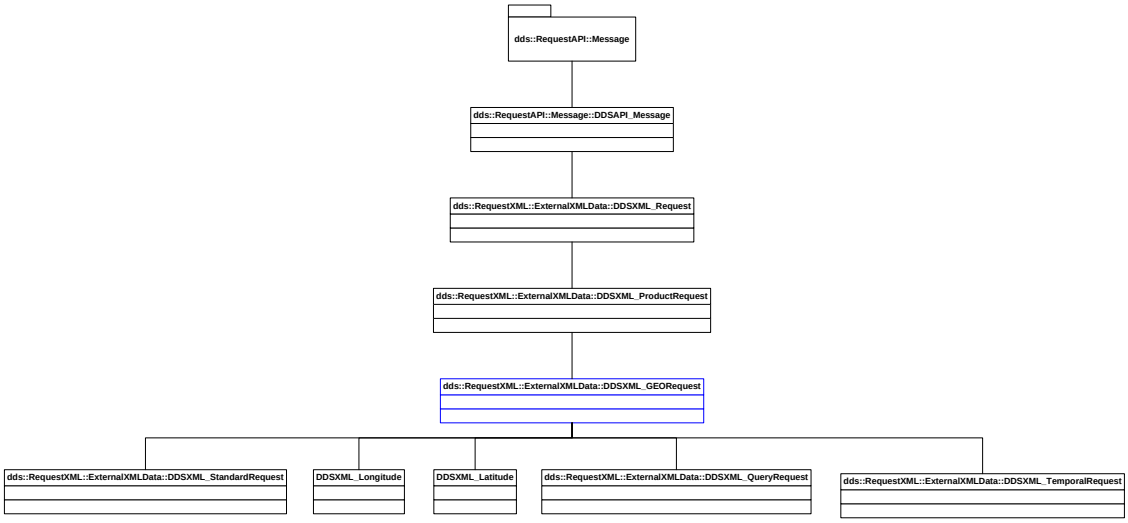

Figure 2.2.10-1, DDSXML_GEORequest Class UML Diagram

2.2.10.1 DDSXML_GEORequest Class Functions

2.2.10.1.1 DDSXML_GEORequest::getAggInterval

long dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getAggInterval (

)

This method returns the value of the AggInterval tag as a long.

Returns:

- long The aggregation interval. If aggregation is 0 then aggregation is off.

2.2.10.1.2 DDSXML_GEORequest::getDelay

long dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getDelay (

)

This method returns the delay for the request in seconds.

Returns:

- long The request's processing delay

2.2.10.1.3 DDSXML_GEORequest::getLowerRightLatitude

DDSXML_Latitude dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getLowerRightLatitude (

)

This method returns the sub-tag value of the LowerRight Latitude tag as a DDSXML_Latitude.

Returns:

- DDSXML_Latitude* A pointer to a DDSXML_Latitude* object that contains the lower right latitude.

2.2.10.1.4 DDSXML_GEORequest::getLowerRightLongitude

DDSXML_Longitude
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getLowerRightLongitude (

)

This method returns the sub-tag value of the LowerRight Longitude tag as a DDSXML_Longitude.

Returns:

- DDSXML_Longitude* A pointer to a DDSXML_Longitude* object that contains the lower right longitude..

2.2.10.1.5 DDSXML_GEORequest::getUpperLeftLatitude

DDSXML_Latitude
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getUpperLeftLatitude (

)

This method returns the sub-tag value of the UpperLeft Latitude tag as a DDSXML_Latitude.

Returns:

- DDSXML_Latitude* A pointer to a DDSXML_Latitude* object that contains the upper left latitude.

2.2.10.1.6 DDSXML_GEORequest::getUpperLeftLongitude

DDSXML_Longitude
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getUpperLeftLongitude (

)

This method returns the sub-tag value of the UpperLeft Longitude tag as a DDSXML_Longitude.

Returns:

- DDSXML_Longitude* A pointer to a DDSXML_Longitude* object that contains the upper left longitude.

2.2.10.1.7 DDSXML_GEORequest::getStartOrbitRevolution

int dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getStartOrbitRevolution (

)

This method returns the starting orbit revolution for the request.

Returns:

- int The orbit revolution number.

2.2.10.1.8 DDSXML_GEORequest::getEndOrbitRevolution

int dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getEndOrbitRevolution (

)

This method returns the last orbit revolution for the request.

Returns:

- int The orbit revolution number.

2.2.10.1.9 DDSXML_GEORequest::getRepaired

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getRepaired (

)

This method returns the value of the Repaired tag as a boolean.

Returns:

- boolean The value of the repaired status flag.

2.2.10.1.10 DDSXML_GEORequest::getPackageState

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getPackageState (

)

This method returns the package state of the request.

Returns:

- boolean The package state of the request.

2.2.10.1.11 DDSXML_GEORequest::isGeospatial

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::isGeospatial (

)

This method returns the status of the geospatial subset.

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

Returns:

- boolean true = The geospatial subset is on false= The geospatial subset is off

2.2.10.1.12 DDSXML_GEORequest::setAggInterval

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setAggInterval (

long aggInterval

)

This method sets the AggInterval tag to the value specified by the aggInterval parameter.

Parameters:

- aggInterval The aggregation interval in microseconds Range: MIN_AGGRINTERVAL >= aggInterval >= MAX_AGGRINTERVAL

Returns:

- boolean - True if an error occurred in executing the command

2.2.10.1.13 DDSXML_GEORequest::setDelay

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setDelay (

long delay

)

This method sets the delay tag to the value specified by the aggInterval parameter.

Parameters:

- delay The delay interval in microseconds Range: MIN_AGGRINTERVAL >= aggInterval >= MAX_AGGRINTERVAL

Returns:

- boolean - True if an error occurred in executing the command

2.2.10.1.14 DDSXML_GEORequest::setGeospatial

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setGeospatial (

boolean geospatialFlag

)

This method sets the SET attribute to the value specified by the activity parameter.

Parameters:

- geospatialFlag is the GeoSpatialSubset active

Returns:

- boolean - True if OK, else false

2.2.10.1.15 DDSXML_GEORequest::setLowerRightLatitude

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setLowerRightLatitude
(

boolean isNegative,

int degrees,

int minutes,

float seconds

)

This method sets the Lower Right Latitude. getGeospatialUsedFlag() must return true for this field to be valid. setGeospatial() must be set to true for this field to be valid. getLowerRightLatitudeUsedFlag() must return true for this field to be valid.

This method will take a Latitude value that is -90 ≤ deg ≤ 90.

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the latitude or longitude as an integer.
- minutes The minutes value for the latitude or longitude as an integer.
- seconds The seconds value for the latitude or longitude as a float.

Returns:

- boolean - True if OK, else false

2.2.10.1.16 DDSXML_GEORequest::setLowerRightLongitude

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setLowerRightLongitude (

boolean isNegative,

int degrees,

int minutes,

float seconds

)

This method sets the Lower Right Longitude. getGeospatialUsedFlag() must return true for this field to be valid. setGeospatial() must be set to true for this field to be valid. getLowerRightLongitudeUsedFlag() must return true for this field to be valid.

This method will take a Lat Long value that is 179 < deg < 360 and subtract 360 degrees from it, to put it in the -180 to +179 range. This will make this lat long value go from a 0 to 359 positioning system to the DDS server standard system of -180 to +179 (where zero in both systems are the same place).

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the latitude or longitude as an integer.
- minutes The minutes value for the latitude or longitude as an integer.
- seconds The seconds value for the latitude or longitude as a float.

Returns:

- boolean - True if OK, else false

2.2.10.1.17 DDSXML_GEORequest::setStartOrbitRevolution

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setStartOrbitRevolution (

int orbitRevolution

)

This method sets the start orbit revolution for the Request. The value, passed in as orbitRevolution, is used as the new value.

Parameters:

- orbitRevolution The new orbit revolution for the request. Range: MIN_ORBITREVOLUTION >= orbitRevolution

Returns:

- boolean - True if an error occurred in executing the command

2.2.10.1.18 DDSXML_GEORequest::setUpperLeftLatitude

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setUpperLeftLatitude (

boolean isNegative,

int degrees,

int minutes,

float seconds

)

This method sets the Upper Left Latitude getGeospatialUsedFlag() must return true for this field to be valid. setGeospatial() must be set to true for this field to be valid. getUpperLeftLatitudeUsedFlag() must return true for this field to be valid.

This method will take a Latitude value that is $-90 \leq deg \leq 90$.

Parameters:

- isNegative Specifies if the LatLong is negative

- degrees The degrees value for the latitude or longitude as an integer.
- minutes The minutes value for the latitude or longitude as an integer.
- seconds The seconds value for the latitude or longitude as a float.

Returns:

- boolean - True if OK, else false

2.2.10.1.19 DDSXML_GEORequest::setUpperLeftLongitude

```
boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setUpperLeftLongitude
(
    boolean isNegative,
    int degrees,
    int minutes,
    float seconds
  )
```

This method sets the Upper Left Longitude. getGeospatialUsedFlag() must return true for this field to be valid. setGeospatial() must be set to true for this field to be valid. getUpperLeftLongitudeUsedFlag() must return true for this field to be valid.

This method will take a Lat Long value that is 179 < deg < 360 and subtract 360 degrees from it, to put it in the -180 to +179 range. This will make this lat long value go from a 0 to 359 positioning system to the DDS server standard system of -180 to +179 (where zero in both systems are the same place).

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the latitude or longitude as an integer.
- minutes The minutes value for the latitude or longitude as an integer.
- seconds The seconds value for the latitude or longitude as a float.

Returns:

- boolean - True if OK, else false

2.2.10.1.20 DDSXML_GEORequest::setEndOrbitRevolution

```
boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setEndOrbitRevolution
(
    int orbitRevolution
  )
```

This method sets the end orbit revolution for the Request. The value, passed in as orbitRevolution, is used as the new value.

Parameters:

- orbitRevolution The new orbit revolution for the request. Range: MIN_ORBITREVOLUTION >= orbitRevolution

Returns:

- boolean - True if an error occurred in executing the command

2.2.10.1.21 DDSXML_GEORequest::setRepaired

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setRepaired (

boolean repaired

)

This method sets the Repaired tag to the value specified by the repaired parameter.

Parameters:

- repaired The new repaired state for the request.

Returns:

- boolean - True if an error occurred in executing the command

2.2.10.1.22 DDSXML_GEORequest::setPackageState

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setPackageState (

boolean packageState

)

Sets the state of each package in the request to the value passed in.

Parameters:

- packageState - The state of the package.

Returns:

- boolean - True if OK, else false

2.2.10.1.23 DDSXML_GEORequest::getAggIntervalUsedFlag

boolean dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getAggIntervalUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.24 DDSXML_GEORequest::getDelayUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getDelayUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.25 DDSXML_GEORequest::getGeospatialUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getGeospatialUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.26 DDSXML_GEORequest::getLowerRightLatitudeUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getLowerRightLatitude UsedFlag (

)

This method returns the value of the used flag. True is returned if the value is used in this request type. Caller owns the object returned

Returns:

- boolean True if used in this request type.

2.2.10.1.27 DDSXML_GEORequest::getLowerRightLongitudeUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getLowerRightLongitu deUsedFlag (

)

This method returns the value of the used flag. True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.28 DDSXML_GEORequest::getUpperLeftLatitudeUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getUpperLeftLatitudeUsedFlag (

)

This method returns the value of the used flag. True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.29 DDSXML_GEORequest::getUpperLeftLongitudeUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getUpperLeftLongitudeUsedFlag (

)

This method returns the value of the used flag. True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.30 DDSXML_GEORequest::getStartOrbitRevolutionUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getStartOrbitRevolutionUsedFlag (

)

This method returns the value of the used flag. True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.31 DDSXML_GEORequest::getEndOrbitRevolutionUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getEndOrbitRevolutionUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.32 DDSXML_GEORequest::getRepairedUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getRepairedUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.33 DDSXML_GEORequest::getPackageStateUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getPackageStateUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.10.1.34 DDSXML_GEORequest::getOrbitIDEnabled

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getOrbitIDEnabled (

)

This method returns the value of the flag

Returns:

- boolean True if enabled in this request.

2.2.10.1.35 DDSXML_GEORequest::setOrbitIDEnabled

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setOrbitIDEnabled (

boolean flag

)

This method sets the value of the flag

Parameters:

- flag True if this is enabled

Returns:

- boolean - True if OK, else false

2.2.10.1.36 DDSXML_GEORequest::getAggregationEnabled

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getAggregationEnabled (

)

This method returns the value of the flag

Returns:

- boolean True if enabled in this request.

2.2.10.1.37 DDSXML_GEORequest::setAggregationEnabled

boolean
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::setAggregationEnabled (

boolean flag

)

This method sets the value of the flag

Parameters:

- flag True if this is enabled

Returns:

- boolean - True if OK, else false

2.2.10.1.38 DDSXML_GEORequest::getMinimumDelay

long
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getMinimumDelay (

)

This method returns the MinimumDelay

Returns:

- long The MinimumDelay

2.2.10.1.39 DDSXML_GEORequest::getMaximumDelay

long
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getMaximumDelay (

)

This method returns the MaximumDelay

Returns:

- long The MaximumDelay

2.2.10.1.40 DDSXML_GEORequest::getMinimumAggregationInterval

long
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getMinimumAggregationInterval (

)

This method returns the MinimumAggregationInterval

Returns:

- long The MinimumAggregationInterval

2.2.10.1.41 DDSXML_GEORequest::getMaximumAggregationInterval

long
dds::RequestXML::ExternalXMLData::DDSXML_GEORequest::getMaximumAggregationInterval (

)

This method returns the MaximumAggregationInterval

Returns:

- long The MaximumAggregationInterval

2.2.11 DDSXML_ImplementationRequestTypesEnum Class Reference

This method returns the implementation type of the Request.

The Class diagram representing the DDSXML_ImplementationRequestTypesEnum Class is provided in Figure 2.2.11.-1, DDSXML_ImplementationRequestTypesEnum UML Diagram.

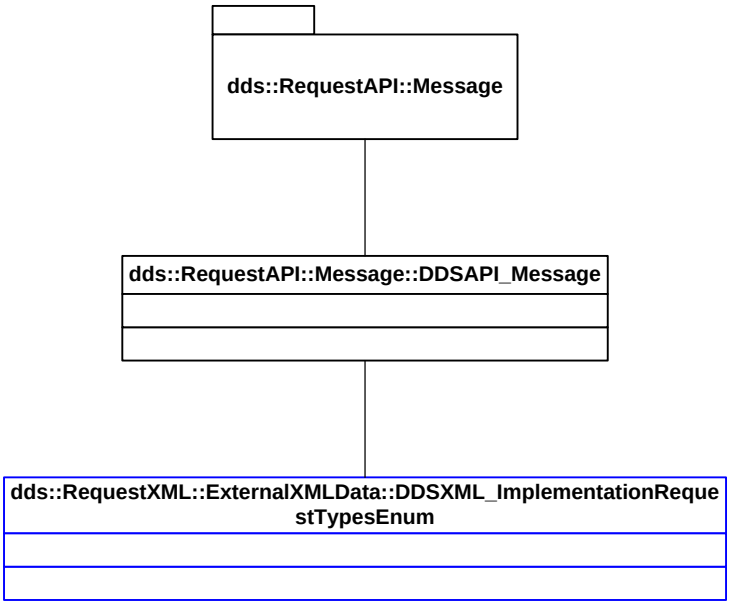

Figure 2.2.11-1, DDSXML_ImplementationRequestTypesEnum Class UML Diagram

2.2.11.1 DDSXML_ImplementationRequestTypesEnum Class Attributes

- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::UNKNOWN_IMPL_REQUEST_TYPE = 0 [static] – Unknown Implementation Type
- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::CATALOG_IMPL_REQUEST_TYPE = 1 [static] – Catalog Implementation Type
- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::DQN_IMPL_REQUEST_TYPE = 2 [static] – DQN Implementation Type
- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::PERIODIC_IMPL_REQUEST_TYPE = 3 [static] – Periodic Implementation Type
- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::STANDARD_IMPL_REQUEST_TYPE = 4 [static] – Standard Implementation Type
- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypes

Enum::QUERY_IMPL_REQUEST_TYPE = 5 [static] – Query Implementation Type

- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::TEMPORAL_IMPL_REQUEST_TYPE = 6 [static] – Temporal Implementation Type
- final int dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::DEFAULT_IMPL_REQUEST_TYPE = 7 [static] – Default Implementation Type

2.2.11.2 DDSXML_ImplementationRequestTypesEnum Class Enumerations

- DDSXML_ImplementationRequestTypesEnum::UNKNOWN_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("UNKNOWN_IMPL_REQUEST_TYPE",UNKNOWN_IMPL_REQUEST_TYPE) – Unknown Implementation Type Enum
- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::CATALOG_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("CATALOG_IMPL_REQUEST_TYPE",CATALOG_IMPL_REQUEST_TYPE) – Catalog Implementation Type Enum
- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::DQN_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("DQN_IMPL_REQUEST_TYPE", DQN_IMPL_REQUEST_TYPE) – DQN Implementation Type Enum
- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::PERIODIC_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("PERIODIC_IMPL_REQUEST_TYPE", PERIODIC_IMPL_REQUEST_TYPE) – Periodic Implementation Type Enum
- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::STANDARD_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("STANDARD_IMPL_REQUEST_TYPE", STANDARD_IMPL_REQUEST_TYPE) – Standard Implementation Type Enum

- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::QUERY_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("QUERY_IMPL_REQUEST_TYPE", QUERY_IMPL_REQUEST_TYPE) – Query Implementation Type Enum
- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::TEMPORAL_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("TEMPORAL_IMPL_REQUEST_TYPE", TEMPORAL_IMPL_REQUEST_TYPE) – Temporal Implementation Type Enum
- DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::DEFAULT_IMPL_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_ImplementationRequestTypesEnum("DEFAULT_IMPL_REQUEST_TYPE", DEFAULT_IMPL_REQUEST_TYPE) – Default Implementation Type Enum

2.2.11.3 DDSXML_ImplementationRequestTypesEnum Class Functions

2.2.11.3.1 DDSXML_ImplementationRequestTypesEnum::findByName

static DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_IMPL_REQUEST_TYPE

Parameters:

- name The string representing an enums name

Returns:

- DDSXML_ImplementationRequestTypesEnum object if found else UNKNOWN_REQUEST_TYPE

2.2.11.3.2 DDSXML_ImplementationRequestTypesEnum::findByValue

static DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_ImplementationRequestTypesEnum::findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_IMPL_REQUEST_TYPE

Parameters:

- value The int representing an enums value

Returns:

- DDSXML_ImplementationRequestTypesEnum object if found else UNKNOWN_IMPL_REQUEST_TYPE

2.2.12 DDSXML_PeriodicRequest Class Reference

This is the XML data class for the Catalog. This class is responsible for storing and maintaining the state of a request in the system.

The Class diagram representing the DDSXML_PeriodicRequest Class is provided in Figure 2.2.12.-1, DDSXML_PeriodicRequest UML Diagram.

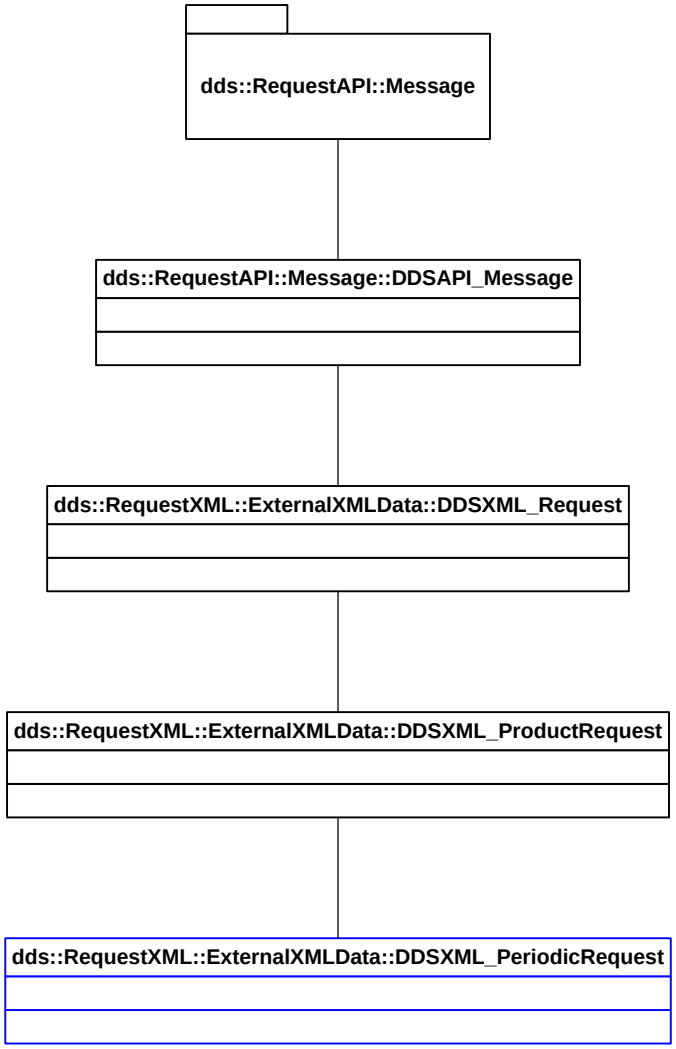

Figure 2.2.12-1, DDSXML_PeriodicRequest Class UML Diagram

2.2.12.1 DDSXML_PeriodicRequest Class Functions

2.2.12.1.1 DDSXML_PeriodicRequest::getDays

int dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::getDays (

)

This method returns the value of the days as an int

Returns:

- int The days for the request.

2.2.12.1.2 DDSXML_PeriodicRequest::setDays

boolean dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::setDays (

int days

)

This method sets the days to the value specified by the parameter.

Parameters:

- days The days for the request.

Returns:

- boolean - True if Ok, else error

2.2.12.1.3 DDSXML_PeriodicRequest::getHours

int dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::getHours (

)

This method returns the value of the Hours as an int

Returns:

- int The Hours for the request.

2.2.12.1.4 DDSXML_PeriodicRequest::setHours

boolean dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::setHours (

int hours

)

This method sets the Hours to the value specified by the parameter.

Parameters:

- hours The Hours for the request.

Returns:

- boolean - True if Ok, else error

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

2.2.12.1.5 DDSXML_PeriodicRequest::getMinutes

int dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::getMinutes (

)

This method returns the value of the Minutes as an int

Returns:

- int The Minutes for the request.

2.2.12.1.6 DDSXML_PeriodicRequest::setMinutes

boolean dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::setMinutes (

int minutes

)

This method sets the Minutes to the value specified by the parameter.

Parameters:

- minutes The minutes for the request.

Returns:

- boolean - True if Ok, else error

2.2.12.1.7 DDSXML_PeriodicRequest::getSeconds

int dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::getSeconds (

)

This method returns the value of the Seconds as an int

Returns:

- int The Seconds for the request.

2.2.12.1.8 DDSXML_PeriodicRequest::setSeconds

boolean dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::setSeconds (

int seconds

)

This method sets the Seconds to the value specified by the parameter.

Parameters:

- seconds The Seconds for the request.

Returns:

- boolean - True if Ok, else error

2.2.12.1.9 DDSXML_PeriodicRequest::getPeriodicityEnabled

boolean
dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::getPeriodicityEnabled (

)

This method returns the value of the enabled flag True is returned if the value is enabled in this request type.

Returns:

- boolean True if enabled in this request type.

2.2.12.1.10 DDSXML_PeriodicRequest::getPeriodicityUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::getPeriodicityUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.12.1.11 DDSXML_PeriodicRequest::setPeriodicityEnabled

boolean
dds::RequestXML::ExternalXMLData::DDSXML_PeriodicRequest::setPeriodicityEnabled (

boolean flag

)

This method sets the value of the enabled flag True is enabled in this request type.

Parameters:

- flag True if enabled for this request.

Returns:

- boolean True if set, else false.

2.2.13 DDSXML_ProductRequest Class Reference

This is the XML data class for the Product Request. This class is responsible for storing and maintaining the state of a catalog in the system.

The Class diagram representing the DDSXML_ProductRequest Class is provided in Figure 2.2.13.-1, DDSXML_ProductRequest UML Diagram.

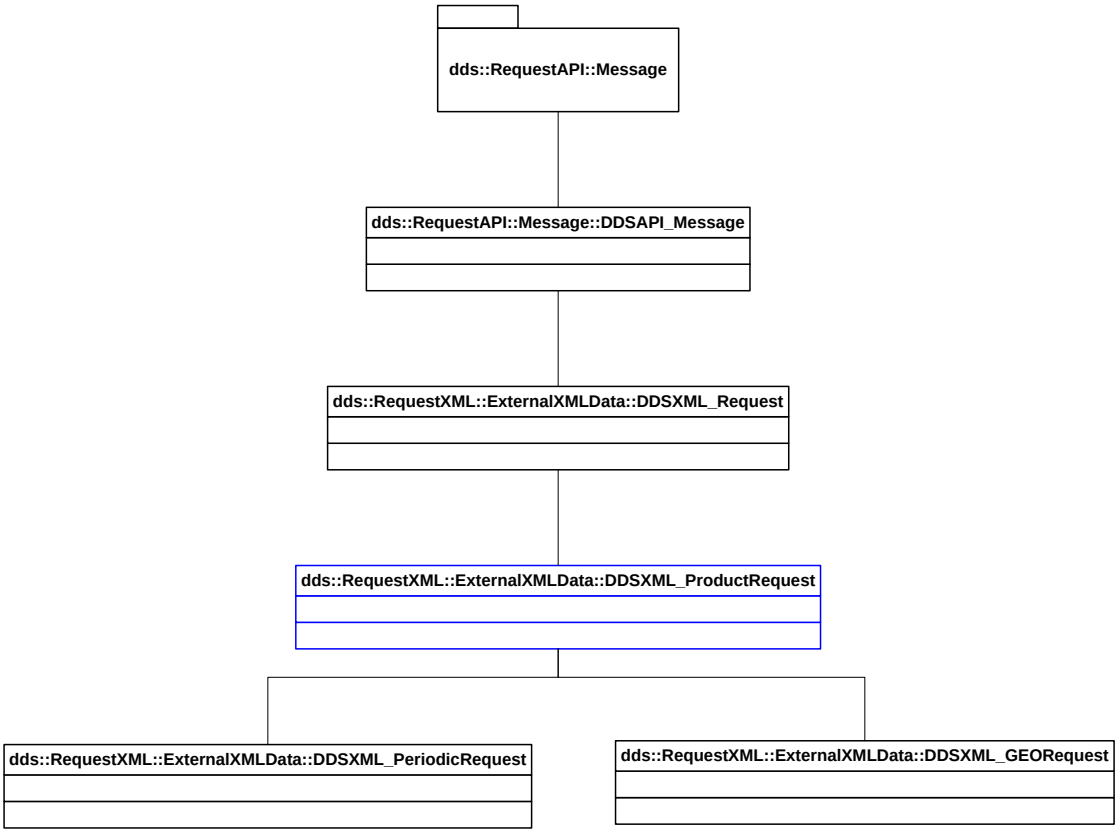

Figure 2.2.13-1, DDSXML_ProductRequest Class UML Diagram

2.2.13.1 DDSXML_ProductRequest Class Functions

2.2.13.1.1 DDSXML_ProductRequest::addDataProduct

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::addDataProduct (

String dataProductID

)

This method adds a dataProduct to the existing Request XML. The dataProduct information is set according to the DataProduct object parameter. The NumberOf DataProducts is also incremented by this method.

Parameters:

- dataProductID The user defined dataProduct ID to be added to the request.

Returns:

- boolean - True if an error occurred in executing the command

2.2.13.1.2 DDSXML_ProductRequest::getDataProducts

Vector
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getDataProducts (

)

This method returns a vector of DataProductID objects that represents all dataProducts ID's currently entered into the request.

Returns:

- Vector The vector of DataProductID objects contained in the request.

2.2.13.1.3 DDSXML_ProductRequest::getNumberOfDataProducts

int
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getNumberOfDataProducts (

)

This method returns the number of dataProducts that currently exist in the request.

Returns:

- int The number of dataProducts in the request.

2.2.13.1.4 DDSXML_ProductRequest::removeDataProduct

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::removeDataProduct (

String dataProductID

)

This method removes a user defined dataProduct from the request.

Parameters:

- dataProductID The user defined dataProduct ID for the dataProduct to be removed from the request or the dataProduct ID was not found.

Returns:

- boolean - True if an error occurred in executing the command

2.2.13.1.5 DDSXML_ProductRequest::dataProductExists

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::dataProductExists (

String dataProductID

)

This method check to see if a DataProduct ID already exists in the request.

Parameters:

- dataProductID The string that contains the dataProduct ID

Returns:

- boolean true - if the DataProduct exists false - if the DataProduct does not exist

2.2.13.1.6 DDSXML_ProductRequest::getStartTime

long dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getStartTime (

)

This method returns the value of the start time as a long.This is for the start time for the request to start looking for data.

Returns:

- long The next Execution Time for the request.

2.2.13.1.7 DDSXML_ProductRequest::getStartTimeAsDate

Date dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getStartTimeAsDate (

)

This method returns the value of the start time as a Date object. This is for the start time for the request to start looking for data.

Returns:

- Date The next Execution Time for the request.

2.2.13.1.8 DDSXML_ProductRequest::setStartTime

boolean dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::setStartTime (

long startTime

)

This method sets the start time to the value specified by the parameter. This is for the start time for the request to start looking for data.

Parameters:

- startTime The start Time for the request.

Returns:

- boolean - True if Ok, else error

2.2.13.1.9 DDSXML_ProductRequest::setStartTimeFromDate

boolean dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::setStartTimeFromDate (

Date startTime

)

This method sets the start time to the value specified by the parameter. This is for the start time for the request to start looking for data.

Parameters:

- startTime The start Time for the request.

Returns:

- boolean - True if Ok, else error

2.2.13.1.10 DDSXML_ProductRequest::getEndTime

long dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getEndTime (

)

This method returns the value of the end time as a long. This is for the end time for the request to stop looking for data.

Returns:

- long The end Time for the request.

2.2.13.1.11 DDSXML_ProductRequest::getEndTimeAsDate

Date dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getEndTimeAsDate (

)

This method returns the value of the end time as a Date object This is for the end time for the request to stop looking for data.

Returns:

- Date The end Time for the request.

2.2.13.1.12 DDSXML_ProductRequest::setEndTime

boolean dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::setEndTime (

long endTime

)

This method sets the end time to the value specified by the parameter. This is for the end time for the request to stop looking for data.

Parameters:

- endTime The end Time for the request.

Returns:

- boolean - True if Ok, else error

2.2.13.1.13 DDSXML_ProductRequest::setEndTimeFromDate

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::setEndTimeFromDate (

Date endTime

)

This method sets the end time to the value specified by the parameter. This is for the end time for the request to stop looking for data.

Parameters:

- endTime The end Time for the request.

Returns:

- boolean - True if Ok, else error

2.2.13.1.14 DDSXML_ProductRequest::getStartTimeUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getStartTimeUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.13.1.15 DDSXML_ProductRequest::getDataProductUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getDataProductUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.13.1.16 DDSXML_ProductRequest::getEndTimeUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getEndTimeUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.13.1.17 DDSXML_ProductRequest::getMinimumStartTime

long
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getMinimumStartTime (

)

This method returns the Minimum Start Time

Returns:

- long The Minimum Start Time in IET (microseconds).

2.2.13.1.18 DDSXML_ProductRequest::getMaximumStartTime

long
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getMaximumStartTime (

)

This method returns the Maximum Start Time

Returns:

- long The Maximum Start Time in IET (microseconds).

2.2.13.1.19 DDSXML_ProductRequest::getMaximumDataProducts

int
dds::RequestXML::ExternalXMLData::DDSXML_ProductRequest::getMaximumDataProducts (

)

This method returns the Maximum Data Products

Returns:

- int The Maximum Data Products

2.2.14 DDSXML_QueryRequest Class Reference

This is the XML data class for the Catalog. This class is responsible for storing and maintaining the state of a request in the system.

The Class diagram representing the DDSXML_QueryRequest Class is provided in Figure 2.2.14.-1, DDSXML_QueryRequest UML Diagram.

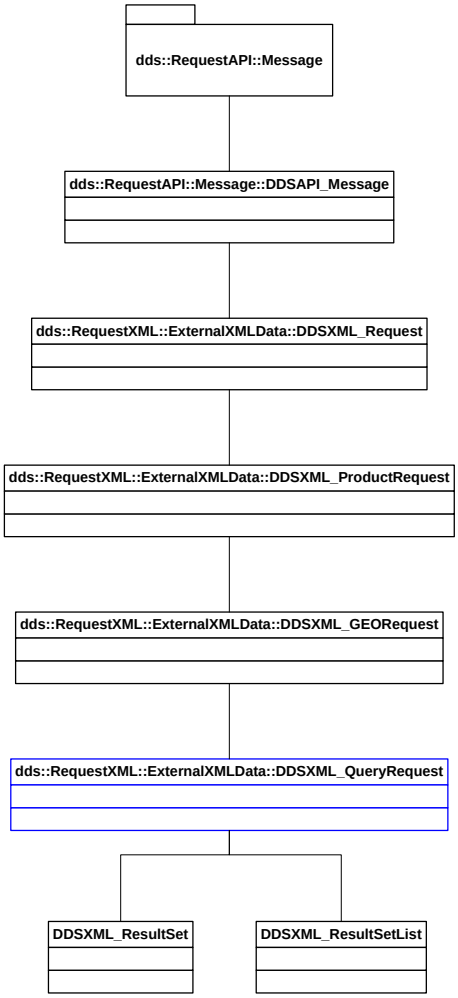

Figure 2.2.14-1, DDSXML_QueryRequest Class UML Diagram

2.2.14.1 DDSXML_QueryRequest Class Functions

2.2.14.1.1 DDSXML_QueryRequest::getResultSets

Vector dds::RequestXML::ExternalXMLData::DDSXML_QueryRequest::getResultSets (

)

This method gets the query results This is for Server/Handler use only Caller owns this

Returns:

- Vector The results of the query.

2.2.14.1.2 DDSXML_QueryRequest::addResultSet

boolean dds::RequestXML::ExternalXMLData::DDSXML_QueryRequest::addResultSet (

DDSXML_ResultSet resultSet

)

This method adds the query results This is for Server/Handler use only

Parameters:

- resultSet The results of the query.

Returns:

- boolean True If OK, false if not

2.2.14.1.3 DDSXML_QueryRequest::getNumberOfResultSets

int
dds::RequestXML::ExternalXMLData::DDSXML_QueryRequest::getNumberOfResultSets (

)

This method returns the number of ResultSets that currently exist in the request.

Returns:

- int The number of ResultSets in the request.

2.2.14.1.4 DDSXML_QueryRequest::getResultSetList

DDSXML_ResultSetList
dds::RequestXML::ExternalXMLData::DDSXML_QueryRequest::getResultSetList (

)

This method gets the query results This is for Server/Handler use only Caller owns this

Returns:

- DDSXML_ResultSetList - The results of the query.

2.2.15 DDSXML_Request Class Reference

This is the base class for all Request types. This class is responsible for storing and maintaining the base state of a request in the system. A Request is made up of a product request type and an implementation request type. Based on the selection made at creation time a specific request will be created and returned. The request returned will be one of the implementation types. This will allow for the validation of the request and the ability to correctly define the progress object. The Data ID (Old Request ID, States, User etc) are in the base DDSXML_UserData class. This allows all data to be treated by the API, Server, and Handler etc. using the same methods for information common to all. This does not include the progress of the Request. (Internal use only) The state of the request can be viewed from the base DDSXML_UserData Class. This does not include the messages of the request.

The Class diagram representing the DDSXML_Request Class is provided in Figure 2.2.15.-1, DDSXML_Request UML Diagram.

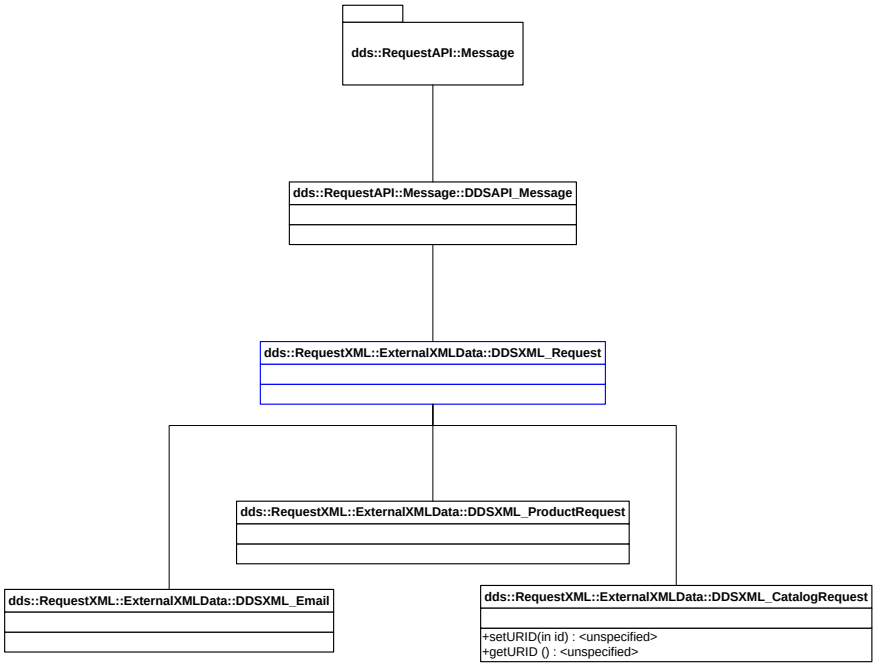

Figure 2.2.15-1, DDSXML_Request Class UML Diagram

2.2.15.1 DDSXML_Request Class Functions

2.2.15.1.1 DDSXML_Request::getRequestType

DDSXML_RequestTypesEnum
dds::RequestXML::ExternalXMLData::DDSXML_Request::getRequestType (

)

This method returns the type of the Request.

Returns:

- DDSXML_RequestTypesEnum the RequestType of the Request or 0 if none

2.2.15.1.2 DDSXML_Request::getRequestTypeName

String
dds::RequestXML::ExternalXMLData::DDSXML_Request::getRequestTypeName (

)

This method returns the type name of the Request.

Returns:

- String The Request type name

2.2.15.1.3 DDSXML_Request::getRequestName

String dds::RequestXML::ExternalXMLData::DDSXML_Request::getRequestName (

)

This method returns the name of the Request.

Returns:

- String the RequestName of the Request

2.2.15.1.4 DDSXML_Request::setRequestName

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::setRequestName (

String requestName

)

This method sets the name of the Request.

Parameters:

- requestName The request name

Returns:

- boolean True if OK, else false

2.2.15.1.5 DDSXML_Request::getRequestMessage

String dds::RequestXML::ExternalXMLData::DDSXML_Request::getRequestMessage (

)

This method returns the message of the Request.

Returns:

- String the Message of the Request

2.2.15.1.6 DDSXML_Request::setRequestMessage

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::setRequestMessage (

String requestMessage

)

This method sets the message of the Request.

Parameters:

- requestMessage The Message of the Request

Returns:

- boolean True if OK, else false

2.2.15.1.7 DDSXML_Request::getRequestImplementationType

DDSXML_ImplementationRequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_Request::getRequestImplementationType (

)

This method returns the implementation type of the Request.

Returns:

- DDSXML_ImplementationRequestTypesEnum The Request Implementation Type of the Request or 0 if none

2.2.15.1.8 DDSXML_Request::getRequestImplementationName

String dds::RequestXML::ExternalXMLData::DDSXML_Request::getRequestImplementationName (

)

This method returns the implementation name of the Request.

Returns:

- String The Request Implementation Type

2.2.15.1.9 Deleted

2.2.15.1.10 DDSXML_Request::isTemplate

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::isTemplate (

)

This method retrieves the template flag for this request. This will allow for any request in the future to be a template.

Returns:

- boolean true = This request is a template false= This request is not a template

2.2.15.1.11 DDSXML_Request::addDestination

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::addDestination (

String destinationID

)

This method adds a destination to the existing Request XML. The destination information is set according to the Destination object parameter. The NumberOf Destinations is also incremented by this method.

Parameters:

- destinationID The user defined destination ID to be added to the request.

Returns:

- boolean - True if OK, else false if not

2.2.15.1.12 DDSXML_Request::getDestinations

Vector dds::RequestXML::ExternalXMLData::DDSXML_Request::getDestinations (

)

This method returns a vector of string objects that represents all destinations ID's currently entered into the request.

Returns:

- Vector The vector of strings that represent all the destination IDs contained in the request.

2.2.15.1.13 DDSXML_Request::getNumberOfDestinations

int
dds::RequestXML::ExternalXMLData::DDSXML_Request::getNumberOfDestinations (

)

This method returns the number of destinations that currently exist in the request.

Returns:

- int The number of destinations in the request.

2.2.15.1.14 DDSXML_Request::removeDestination

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::removeDestination (

String destinationID

)

This method removes a user defined destination from the request.

Parameters:

- destinationID The user defined destination ID for the destination to be removed from the request or the destination ID was not found.

Returns:

- boolean - True if an error occurred in executing the command

2.2.15.1.15 DDSXML_Request::destinationExists

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::destinationExists (

String destinationID

)

This method check to see if a Destination ID already exists in the request.

Parameters:

- destinationID The string that contains the destination ID

Returns:

- boolean true - if the Destination exists false - if the Destination does not exist

2.2.15.1.16 Deleted

2.2.15.1.17 DDSXML_Request::getImplementationTypes

static Vector
dds::RequestXML::ExternalXMLData::DDSXML_Request::getImplementationTypes (

DDSXML_RequestTypesEnum requestType

)

This method returns a vector of implementation types that can be used for the Request type passed in. This is static so it can be called without needing an instance of this

Parameters:

- requestType The request type

Returns:

- Vector The vector of strings that represent all the possible DDSXML_ImplementationRequestTypes for a specific request type.

2.2.15.1.18 DDSXML_Request::getSuspendTime

long dds::RequestXML::ExternalXMLData::DDSXML_Request::getSuspendTime (

)

This method returns the value of the Suspend time as a long

Returns:

- long The next Execution Time for the request.

2.2.15.1.19 DDSXML_Request::setSuspendTime

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::setSuspendTime (

long suspendTime

)

This method sets the Suspend Time to the value specified by the parameter. The time is in IET time. (microseconds)

Parameters:

- suspendTime The Suspend Time for the request.

Returns:

- boolean - True if Ok, else error

2.2.15.1.20 DDSXML_Request:: getMinimumSuspendDuration

long dds::RequestXML::ExternalXMLData::DDSXML_Request::
getMinimumSuspendDuration (

)

This method returns the Minimum Suspend Duration as a long

Returns:

- long The Minimum Suspend Duration for the request.

2.2.15.1.21 DDSXML_Request:: getMaximumSuspendDuration

long dds::RequestXML::ExternalXMLData::DDSXML_Request::
getMaximumSuspendDuration (

)

This method returns the Maximum Suspend Duration as a long.

Returns:

- long The Minimum Suspend Duration for the request

2.2.15.1.22 Deleted

2.2.15.1.23 Deleted

2.2.15.1.24 Deleted

2.2.15.1.25 DDSXML_Request::setTemplate

boolean dds::RequestXML::ExternalXMLData::DDSXML_Request::setTemplate (

boolean templateFlag

)

This method sets the template flag for this request. This will allow for any request in the future to be a template.

Parameters:

- templateFlag true = This request is a template false= This request is not a template

Returns:

- boolean true if set ok, false if not

2.2.15.1.26 DDSXML_Request::getProgressFilename

String dds::RequestXML::ExternalXMLData::DDSXML_Request::getProgressFilename (

)

This method gets the name of the progress file.

Returns:

- String The progress file name or empty if none

2.2.15.1.27 DDSXML_Request::getDestinationUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Request::getDestinationUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.15.1.28 Deleted

2.2.15.1.29 DDSXML_Request::getTemplateUsedFlag

boolean
dds::RequestXML::ExternalXMLData::DDSXML_Request::getTemplateUsedFlag (

)

This method returns the value of the used flag True is returned if the value is used in this request type.

Returns:

- boolean True if used in this request type.

2.2.16 DDSXML_RequestTypesEnum Class Reference

This class is defines all valid request types.

The Class diagram representing the DDSXML_RequestTypesEnum Class is provided in Figure 2.2.16.-1, DDSXML_RequestTypesEnum UML Diagram.

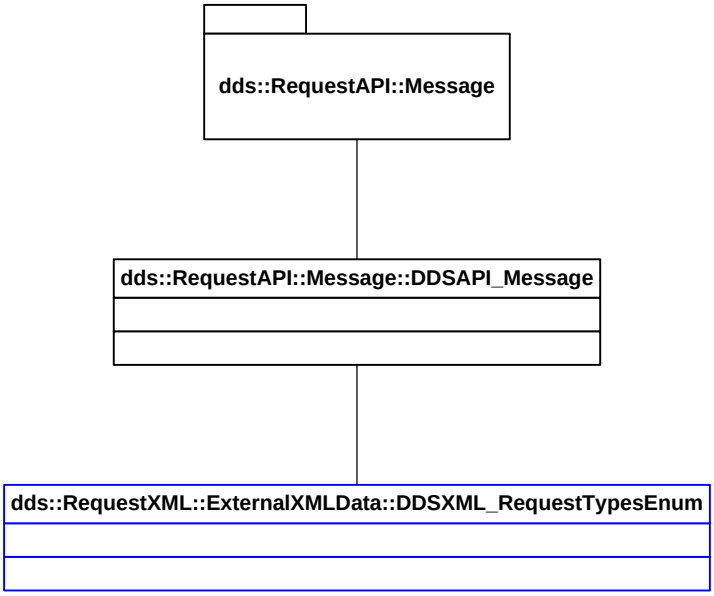

Figure 2.2.16-1, DDSXML_RequestTypesEnum Class UML Diagram

2.2.16.1 DDSXML_RequestTypesEnum Class Attributes

- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::UNKNOWN_REQUEST_TYPE = 0 – Unknown Request Type Enum
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::SDR_EDR_IP_REQUEST_TYPE = 1 – SDR, EDR, IP Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::RDR_REQUEST_TYPE = 2 – RDR Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::ANC_REQUEST_TYPE = 3 – ANC Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::AUX_REQUEST_TYPE = 4 – AUX Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::DQN_REQUEST_TYPE = 5 – DQN Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::PERIODIC_REQUEST_TYPE = 6 – Gridded IP Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::DIARY_REQUEST_TYPE = 7 – Diary Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::CATALOG_REQUEST_TYPE = 8 – Catalog Request Type
- static final int dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::INVALID_REQUEST_TYPE = 9 – Invalid Request Type Enum

2.2.16.2 DDSXML_RequestTypesEnum Class Enumerations

- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::UNKNOWN_REQUEST_TYPE_ENUM [static] – Initial value: new DDSXML_RequestTypesEnum("UNKNOWN_REQUEST_TYPE", UNKNOWN_REQUEST_TYPE) – Unknown Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::SDR_E

DR_IP_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("SDR_EDR_IP_REQUEST_TYPE", SDR_EDR_IP_REQUEST_TYPE) – SDR, EDR, IP Request Type Enum

- static DDSXML_RequestTypesEnum.java.DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::RDR_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("RDR_REQUEST_TYPE", RDR_REQUEST_TYPE) – RDR Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::ANC_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("ANC_REQUEST_TYPE", ANC_REQUEST_TYPE) – ANC Request Type Enum
- DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::AUX_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("AUX_REQUEST_TYPE", AUX_REQUEST_TYPE) – AUX Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::DQN_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("DQN_REQUEST_TYPE", DQN_REQUEST_TYPE) – DQN Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::PERIODIC_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("PERIODIC_REQUEST_TYPE",PERIODIC_REQUEST_TYPE) – Periodic IP Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::DIARY_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("DIARY_REQUEST_TYPE", DIARY_REQUEST_TYPE) – Diary Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::CATALOG_REQUEST_TYPE_ENUM – Initial value: new DDSXML_RequestTypesEnum("CATALOG_REQUEST_TYPE", CATALOG_REQUEST_TYPE) – Catalog Request Type Enum
- static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::INVALID_REQUEST_TYPE_ENUM – Initial value: new

DDSXML_RequestTypesEnum("INVALID_REQUEST_TYPE", INVALID_REQUEST_TYPE) – Invalid Request Type Enum

2.2.16.3 DDSXML_RequestTypesEnum Class Functions

2.2.16.3.1 DDSXML_RequestTypesEnum::findByName

static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_REQUEST_TYPE

Parameters:

- name The string representing an enums name

Returns:

- DDSXML_RequestTypesEnum object if found else UNKNOWN_REQUEST_TYPE

2.2.16.3.2 DDSXML_RequestTypesEnum::findByValue

static DDSXML_RequestTypesEnum dds::RequestXML::ExternalXMLData::DDSXML_RequestTypesEnum::findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_REQUEST_TYPE

Parameters:

- value The int representing an enums value

Returns:

- DDSXML_RequestTypesEnum object if found else UNKNOWN_REQUEST_TYPE

2.2.17 DDSXML_StandardRequest Class Reference

This is the XML data class for the Standard Request. This class is responsible for storing and maintaining the state of a request in the system.

The Class diagram representing the DDSXML_StandardRequest Class is provided in Figure 2.2.17.-1, DDSXML_StandardRequest UML Diagram.

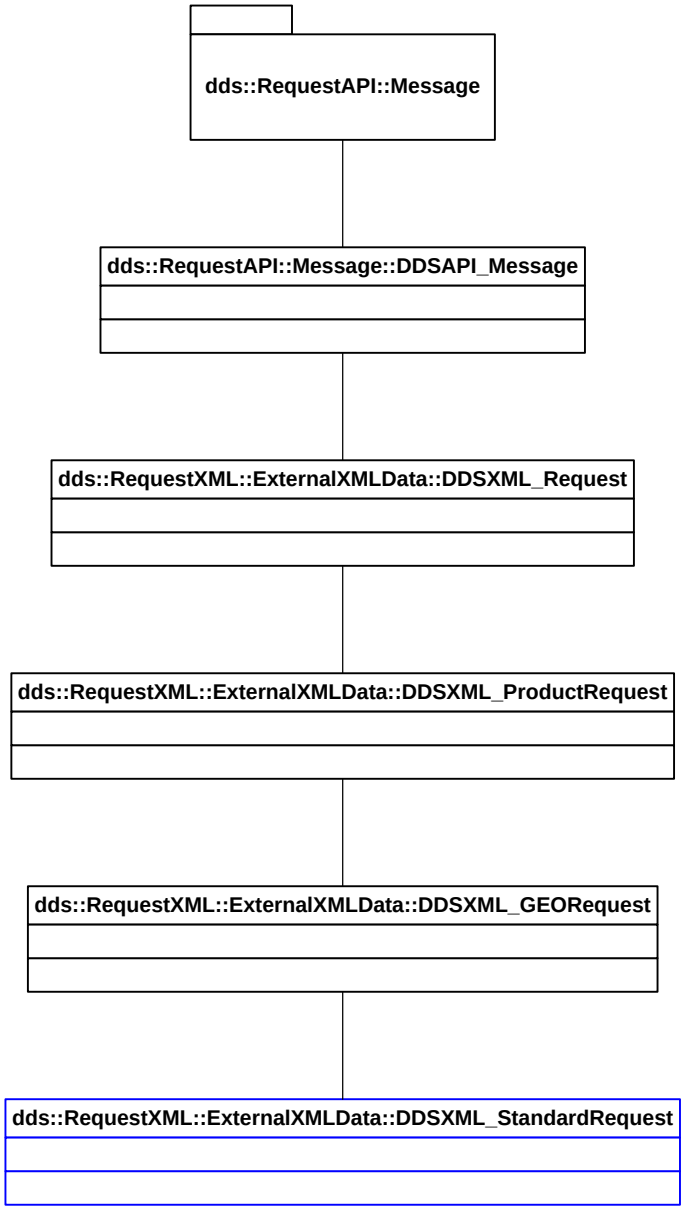

Figure 2.2.17-1, DDSXML_StandardRequest Class UML Diagram

2.2.18 DDSXML_SystemMessage Class Reference

This class describes the current status of the deny data roles in the system. This is only valid in the Java and C++ versions. Only the DDS Java GUI may be used to deny data. Only the DDS Java Client may deny Roles to the C++ Server.

The Class diagram representing the DDSXML_SystemMessage Class is provided in Figure 2.2.18.-1, DDSXML_SystemMessage UML Diagram.

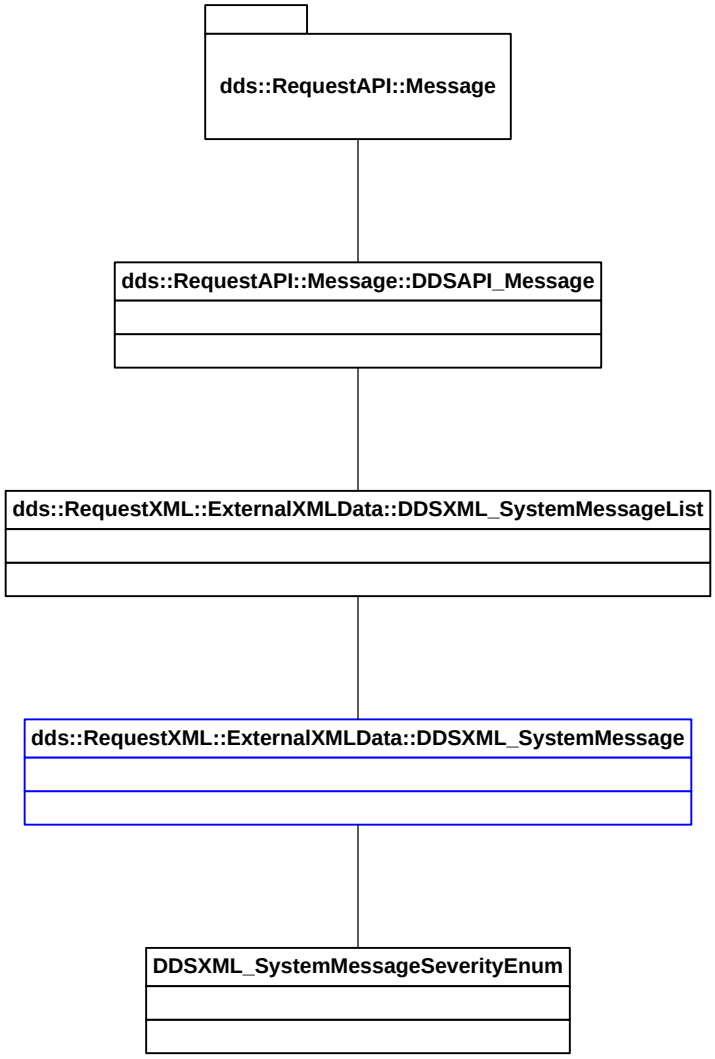

Figure 2.2.18-1, DDSXML_SystemMessage Class UML Diagram

2.2.18.1 DDSXML_SystemMessage Class Functions

2.2.18.1.1 DDSXML_SystemMessage::getMessage

String dds::RequestXML::ExternalXMLData::DDSXML_SystemMessage::getMessage ( )

This method retrieves the message

Returns:

- String The message .

2.2.18.1.2 DDSXML_SystemMessage::setMessage

void dds::RequestXML::ExternalXMLData::DDSXML_SystemMessage::setMessage (String message

)

This method sets the message.

Parameters:

- message String The message .

2.2.18.1.3 DDSXML_SystemMessage::getTimeStamp

long dds::RequestXML::ExternalXMLData::DDSXML_SystemMessage::getTimeStamp (

)

This method retrieves the TimeStamp from the XML.

Returns:

- long The TimeStamp .

2.2.18.1.4 DDSXML_SystemMessage::setTimeStamp

void dds::RequestXML::ExternalXMLData::DDSXML_SystemMessage::setTimeStamp (

long timeStamp

)

This method sets the TimeStamp.

Parameters:

- timeStamp The TimeStamp .

2.2.18.1.5 DDSXML_SystemMessage::setSeverity

void dds::RequestXML::ExternalXMLData::DDSXML_SystemMessage::setSeverity (

DDSXML_SystemMessageSeverityEnum severity

)

This method sets the Severity.

Parameters:

- severity The Severity .

2.2.18.1.6 DDSXML_SystemMessage::getSeverity

DDSXML_SystemMessageSeverityEnum dds::RequestXML::ExternalXMLData::DDSXML_SystemMessage::getSeverity (

)

This method retrieves the Severity from the XML.

Returns:

- DDSXML_SystemMessageSeverityEnum The Severity .

2.2.19 DDSXML_SystemMessageList Class Reference

This class is used to keep a list of system messages. This class is based on the Base XML class. All pointers returned, referenced memory is owned by the API and should not be destroyed by the caller unless noted in the method called.

The Class diagram representing the DDSXML_SystemMessageList Class is provided in Figure 2.2.19.-1, DDSXML_SystemMessageList UML Diagram.

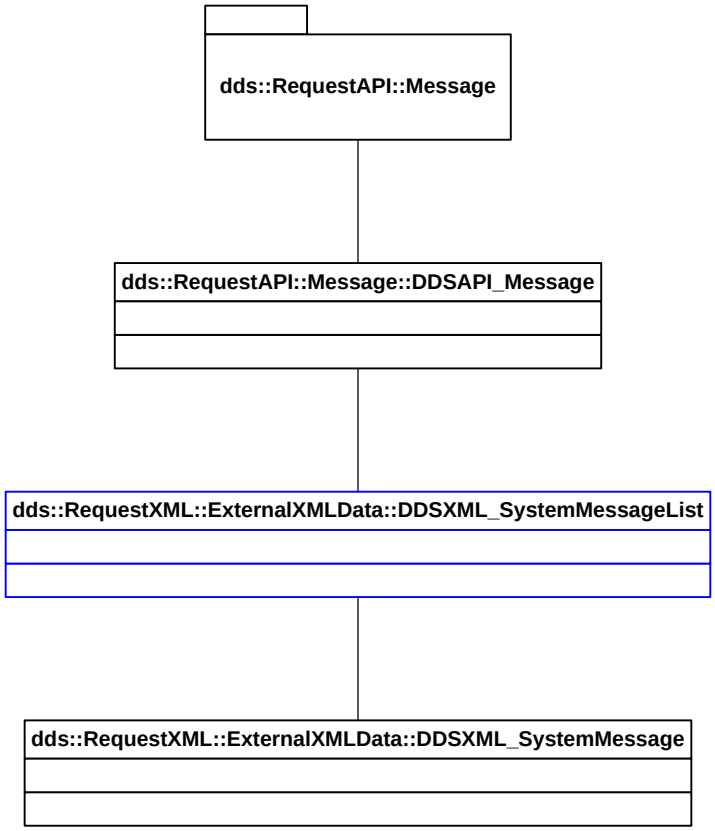

Figure 2.2.19-1, DDSXML_SystemMessageList Class UML Diagram

2.2.19.1 DDSXML_SystemMessageList Class Functions

2.2.19.1.1 DDSXML_SystemMessageList::addSystemMessage

boolean
dds::RequestXML::ExternalXMLData::DDSXML_SystemMessageList::addSystemMessage (

DDSXML_SystemMessage msg

)

This method adds system message to this list.

Parameters:

- msg The system message

2.2.19.1.2 DDSXML_SystemMessageList::getSystemMessages

Vector
dds::RequestXML::ExternalXMLData::DDSXML_SystemMessageList::getSystemMessages (

)

This method returns the system messages.

Returns:

- A vector of the system messages

2.2.20 DDSXML_TemporalRequest Class Reference

This is the XML data class for the Temporal Request. This class is responsible for storing and maintaining the state of a request in the system.

The Class diagram representing the DDSXML_TemporalRequest Class is provided in Figure 2.2.20.-1, DDSXML_TemporalRequest UML Diagram.

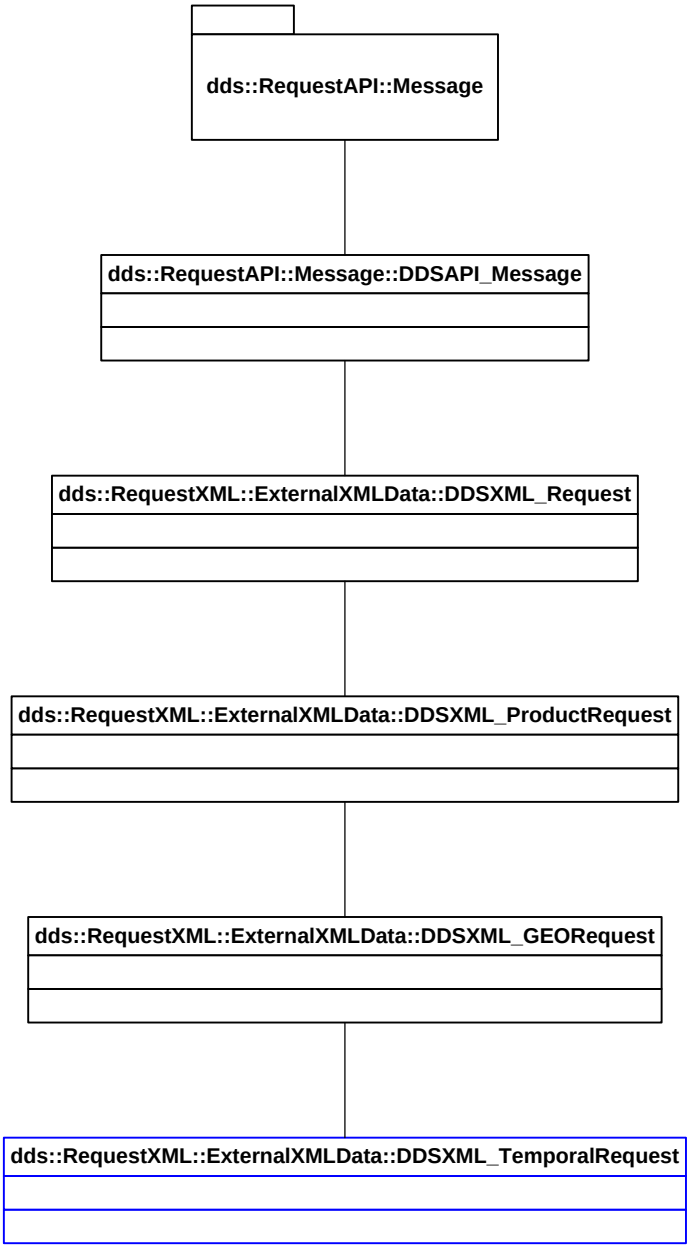

Figure 2.2.20-1, DDSXML_TemporalRequest Class UML Diagram

2.2.20.1 DDSXML_TemporalRequest Class Functions

2.2.20.1.1 DDSXML_TemporalRequest::getTemporalStart

long
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::getTemporalStart (
)

This method returns the temporal start for the request in long. The time is in microseconds since midnight. getTemporalStartUsedFlag() must return true for this field to be valid.

Returns:

- long The request's temporal start

2.2.20.1.2 DDSXML_TemporalRequest::getTemporalDuration

long
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::getTemporalDuration (

)

This method returns the temporal duration for the request in long. The time is in microseconds. getTemporalDurationUsedFlag() must return true for this field to be valid.

Returns:

- long The request's temporal duration

2.2.20.1.3 DDSXML_TemporalRequest::setTemporalStart

boolean
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::setTemporalStart (

long temporalStart

)

This method sets the temporal start for the request in Int64. The time is in microseconds since midnight. This must be between getMinimumTemporalStart() and getMaximumTemporalStart(); getTemporalStartUsedFlag() must return true for this field to be valid.

Parameters:

- temporalStart The request's temporal start

Returns:

- boolean True if set ok

2.2.20.1.4 DDSXML_TemporalRequest::setTemporalDuration

boolean
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::setTemporalDuration (

long duration

)

This method sets the temporal duration for the request in seconds. This must be between getMinimumTemporalDuration() and getMaximumTemporalDuration(); getTemporalDurationUsedFlag() must return true for this field to be valid. The time is in microseconds.

Parameters:

- duration The request's temporal duration

Returns:

- boolean True if set ok

2.2.20.1.5 DDSXML_TemporalRequest::getMinimumTemporalStart

long
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::getMinimumTemporalStart (

)

This method returns the MinimumTemporalStart

Returns:

- long The MinimumTemporalStart

2.2.20.1.6 DDSXML_TemporalRequest::getMaximumTemporalStart

long
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::getMaximumTemporalStart (

)

This method returns the MaximumTemporalStart

Returns:

- long The MaximumTemporalStart

2.2.20.1.7 DDSXML_TemporalRequest::getMinimumTemporalDuration

long
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::getMinimumTemporalDuration (

)

This method returns the MinimumTemporalDuration

Returns:

- long The MinimumTemporalDuration

2.2.20.1.8 DDSXML_TemporalRequest::getMaximumTemporalDuration

long
dds::RequestXML::ExternalXMLData::DDSXML_TemporalRequest::getMaximumTemporalDuration (

)

This method returns the MaximumTemporalDuration

Returns:

- long The MaximumTemporalDuration

2.2.21 DDSXML_User Class Reference

This class is responsible for handling the user XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. The DDS system uses a username and user role as one unique user. All searches are base on the user name and user role as a key.

The Class diagram representing the DDSXML_User Class is provided in Figure 2.2.21.-1, DDSXML_User UML Diagram.

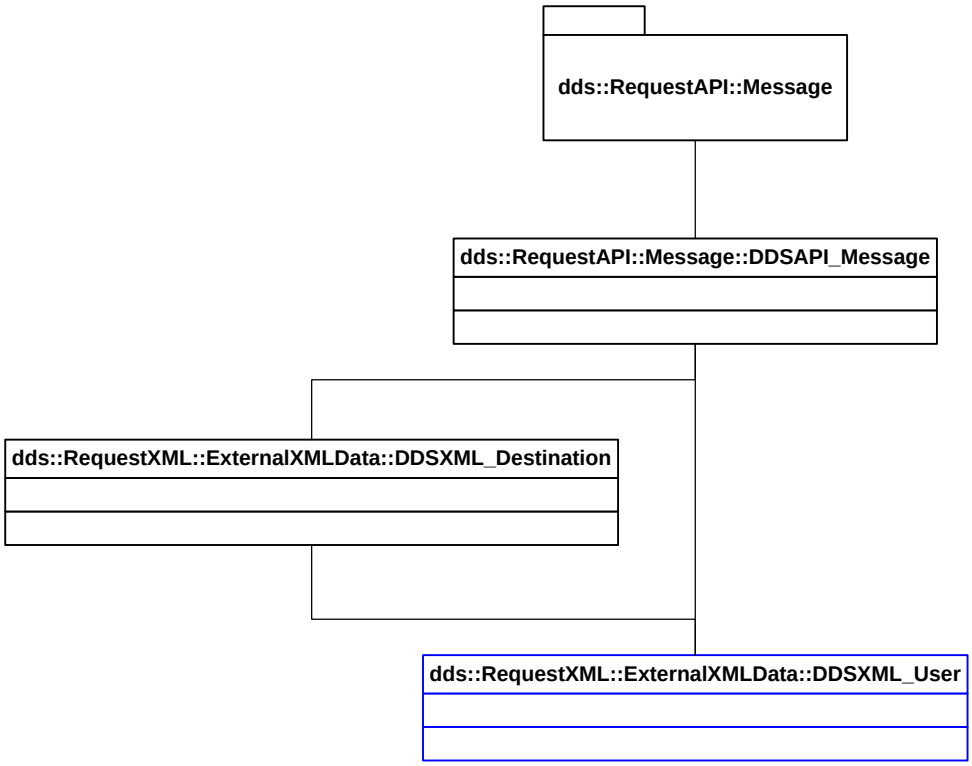

Figure 2.2.21-1, DDSXML_User Class UML Diagram

2.2.21.1 DDSXML_User Class Functions

2.2.21.1.1 DDSXML_User::setUser

boolean dds::RequestXML::ExternalXMLData::DDSXML_User::setUser (

DDSXML_User user

)

This method sets the user.

Parameters:

- user The user.

Returns:

- boolean True if set

2.2.21.1.2 DDSXML_User::getUserName

String dds::RequestXML::ExternalXMLData::DDSXML_User::getUserName (

)

This method retrieves the user name from the message.

Returns:

- String The user name. or empty if not set.

2.2.21.1.3 DDSXML_User::setUserName

boolean dds::RequestXML::ExternalXMLData::DDSXML_User::setUserName (

String name

)

This method sets the users name.

Parameters:

- name The user name. or empty if not set.

Returns:

- boolean True

2.2.21.1.4 DDSXML_User::getUserRole

String dds::RequestXML::ExternalXMLData::DDSXML_User::getUserRole (

)

This method retrieves the user role from the message.

Returns:

- String The user role.or empty if not set.

2.2.21.1.5 DDSXML_User::setUserRole

boolean dds::RequestXML::ExternalXMLData::DDSXML_User::setUserRole (

String role

)

This method sets the users role.

Parameters:

- role String The user role.

Returns:

- boolean True if set

2.2.21.1.6 DDSXML_User::getKey

String dds::RequestXML::ExternalXMLData::DDSXML_User::getKey (

)

This method retrieves this key.

Returns:

- String The key for this user role or empty if not set.

2.2.22 DDSXML_DataShipmentStatesEnum Class Reference

This class is defines all destination states.

The Class diagram representing the DDSXML_DataShipmentStatesEnum Class is provided in Figure 2.2.22.-1, DDSXML_DataShipmentStatesEnum UML Diagram.

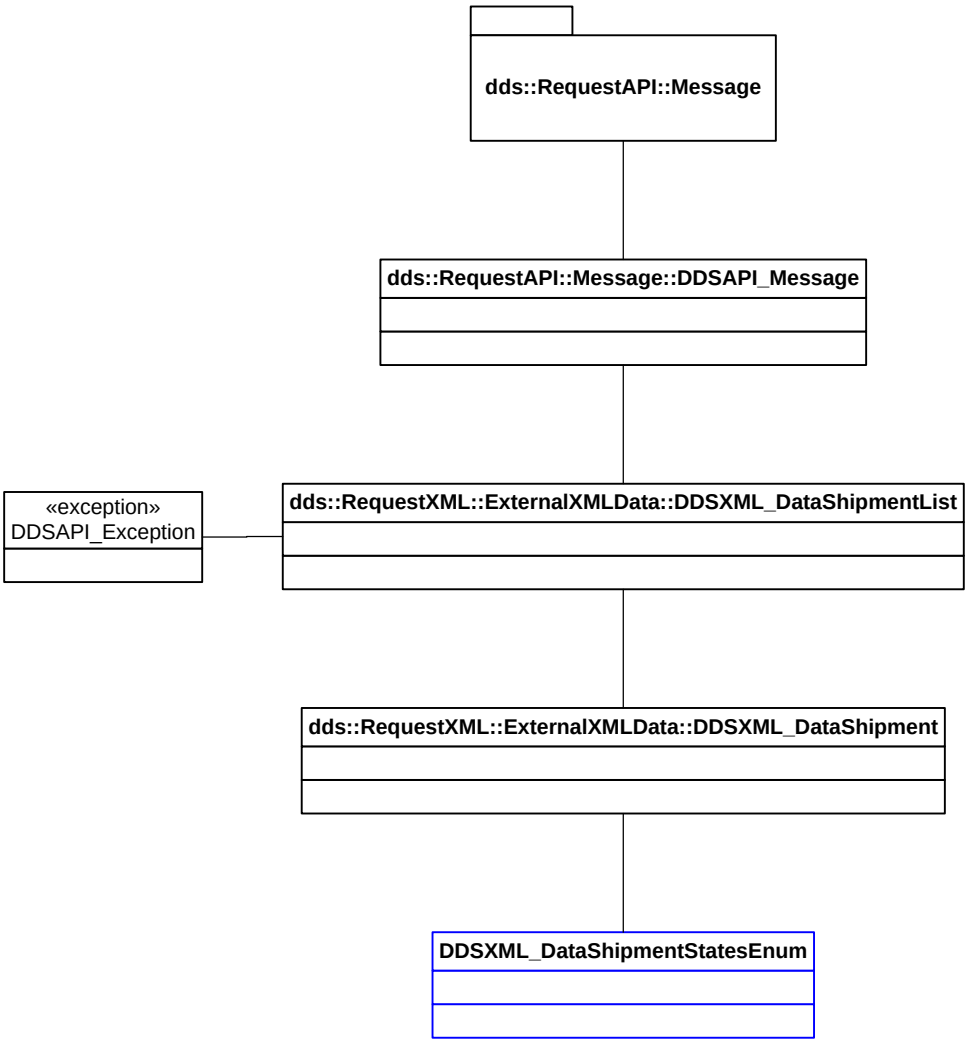

Figure 2.2.22-1, DDSXML_DataShipmentStatesEnum Class UML Diagram

2.2.22.1 DDSXML_DataShipmentStatesEnum Class Attributes

- static final int dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.DATA_SHIPMENT_UNKNOWN = 0
- static final int dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.DATA_SHIPMENT_COMPLETED = 1
- static final int dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.DATA_SHIPMENT_FAILED = 2

2.2.22.2 DDSXML_DataShipmentStatesEnum Class Enumerations

- static DDSXML_DataShipmentStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.DATA_SHIPMENT_UNKNOWN_ENUM – Initial value: new DDSXML_DataShipmentStatesEnum("DATA_SHIPMENT_UNKNOWN", DATA_SHIPMENT_UNKNOWN, " DATA_SHIPMENT_UNKNOWN ")
- static DDSXML_DataShipmentStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.DATA_SHIPMENT_COMPLETED_ENUM – Initial value: new DDSXML_DataShipmentStatesEnum("DATA_SHIPMENT_COMPLETED", DATA_SHIPMENT_COMPLETED, " DATA_SHIPMENT_COMPLETED ")
- static DDSXML_DataShipmentStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.DATA_SHIPMENT_FAILED_ENUM – Initial value: new DDSXML_DataShipmentStatesEnum("DATA_SHIPMENT_FAILED", DATA_SHIPMENT_FAILED, " DATA_SHIPMENT_FAILED ")

2.2.22.3 DDSXML_DataShipmentStatesEnum Class Functions

2.2.22.3.1 DDSXML_DataShipmentStatesEnum.findByName

static DDSXML_DataShipmentStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.findByName(

String name

)

Find the enum that corresponds to the given string. If can't find it just returns UNKNOWN_CMD_ENUM

Parameters:

- name The string representing an enums name

Returns:

- DDSXML_DataShipmentStatesEnum The enum object if found else UNKNOWN_DATA_TYPE

2.2.22.3.2 DDSXML_DataShipmentStatesEnum.findByValue

static DDSXML_DataShipmentStatesEnum
dds.RequestXML.ExternalXMLData.DDSXML_DataShipmentStatesEnum.findByValue
(

int value

)

Find the enum that corresponds to the given string. If can't find it just returns UNKNOWN_CMD_ENUM

Parameters:

- value The int representing an enums value

Returns:

- DDSXML_DataShipmentStatesEnum The enum object if found else UNKNOWN_DATA_TYPE

2.2.23 DDSXML_SystemMessageSeverityEnum Class Reference

This class defines all destination states.

The Class diagram representing the DDSXML_ SystemMessageSeverityEnum Class is provided in Figure 2.2.23.-1, DDSXML_ SystemMessageSeverityEnum UML Diagram.

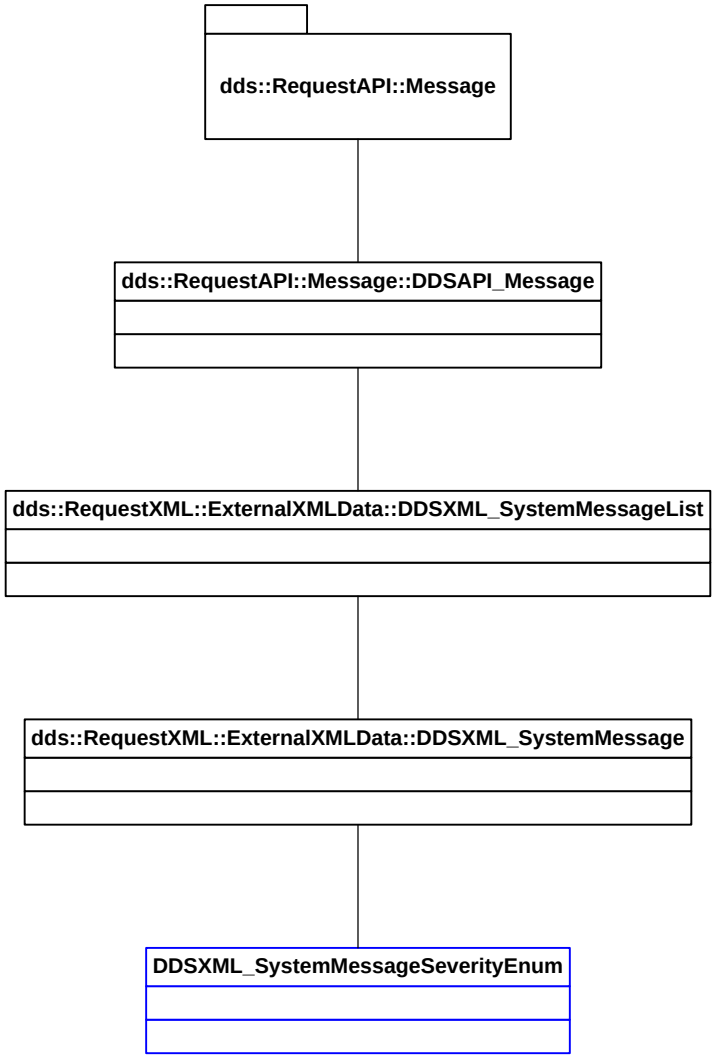

Figure 2.2.23-1, DDSXML_ SystemMessageSeverityEnum Class UML Diagram

2.2.23.1 DDSXML_ SystemMessageSeverityEnum Class Attributes

- static final int dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.ENGINEERING_MSG = 0
- static final int dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.INFORMATION_MSG = 1
- static final int dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.WARNING_MSG = 2
- static final int dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.CRITICAL_MSG = 3

- static final int dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.UNDEFINED_MSG = 4

2.2.23.2 DDSXML_ SystemMessageSeverityEnum Class Enumerations

- static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.ENGINEERING_MSG_ENUM – Initial value: new DDSXML_SystemMessageSeverityEnum("ENGINEERING_MSG", ENGINEERING_MSG)
- static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.INFORMATION_MSG_ENUM – Initial value: new DDSXML_SystemMessageSeverityEnum("INFORMATION_MSG", INFORMATION_MSG)
- static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.WARNING_MSG_ENUM – Initial value: new DDSXML_SystemMessageSeverityEnum("WARNING_MSG", WARNING_MSG)
- static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.CRITICAL_MSG_ENUM – Initial value: new DDSXML_SystemMessageSeverityEnum("CRITICAL_MSG", CRITICAL_MSG)
- static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.UNDEFINED_MSG_ENUM – Initial value: new DDSXML_SystemMessageSeverityEnum("UNDEFINED_MSG", UNDEFINED_MSG)

2.2.23.3 DDSXML_ SystemMessageSeverityEnum Class Functions

2.2.23.3.1 DDSXML_SystemMessageSeverityEnum.findByName

static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_CMD_ENUM

Parameters:

- val The string representing an enums name

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

2.2.23.3.2 DSXML_SystemMessageSeverityEnum.findByValue

static DDSXML_SystemMessageSeverityEnum dds.RequestXML.ExternalXMLData.DDSXML_SystemMessageSeverityEnum.findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just returns UNKNOWN_CMD_ENUM

Parameters:

- val The int representing an enums value

Returns:

- The ensum object if found else UNKNOWN_DATA_TYPE

2.2.24 DDSXML_DestinationTransferTypeEnum Class Reference

This class is defines all destination transfer types.

The Class diagram representing the DDSXML_ DestinationTransferTypeEnum Class is provided in Figure 2.2.24.-1, DDSXML_ DestinationTransferTypeEnum UML Diagram.

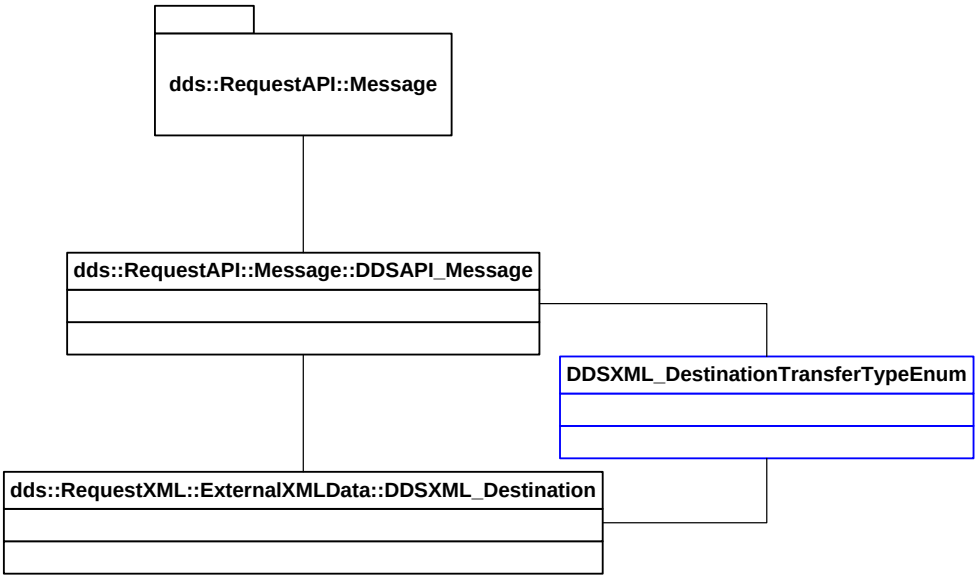

Figure 2.2.24-1, DDSXML_ DestinationTransferTypeEnum Class UML Diagram

2.2.24.1 DDSXML_ DestinationTransferTypeEnum Class Attributes

- static DDSXML_DestinationTransferTypeEnum DESTINATION_TRANSFER_UNKNOWN
- static DDSXML_DestinationTransferTypeEnum DESTINATION_TRANSFER_LOCAL
- static DDSXML_DestinationTransferTypeEnum DESTINATION_TRANSFER_FTP
- static DDSXML_DestinationTransferTypeEnum DESTINATION_TRANSFER_SFTP
- static DDSXML_DestinationTransferTypeEnum DESTINATION_TRANSFER_FTPS

2.2.24.2 DDSXML_ DestinationTransferTypeEnum Class Enumerations

- static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum. DESTINATION_TRANSFER_UNKNOWN – Initial value: new DDSXML_DestinationTransferTypeEnum("DESTINATION_TRANSFER_UNKNOWN", index++)
- static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum. DESTINATION_TRANSFER_LOCAL – Initial value: new DDSXML_DestinationTransferTypeEnum("DESTINATION_TRANSFER_LOCAL", index++)
- static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum. DESTINATION_TRANSFER_FTP – Initial value: new DDSXML_DestinationTransferTypeEnum("DESTINATION_TRANSFER_FTP", index++)
- static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum. DESTINATION_TRANSFER_SFTP – Initial value: new DDSXML_DestinationTransferTypeEnum("DESTINATION_TRANSFER_SFTP", index++)
- static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum. DESTINATION_TRANSFER_FTPS – Initial value: new DDSXML_DestinationTransferTypeEnum("DESTINATION_TRANSFER_FTPS", index++)

2.2.24.3 DDSXML_ DestinationTransferTypeEnum Class Functions

2.2.24.3.1 DDSXML_DestinationTransferTypeEnum.findByName

static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum.findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just returns DESTINATION_TRANSFER_UNKNOWN

Parameters:

- name The string representing an enums name

Returns:

- DDSXML_DestinationTransferTypeEnum The enum object if found else UNKNOWN_DATA_TYPE

2.2.24.3.2 DDSXML_DestinationTransferTypeEnum.findByValue

static DDSXML_DestinationTransferTypeEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationTransferTypeEnum.findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just returns DESTINATION_TRANSFER_UNKNOWN

Parameters:

- value The int representing an enums value

Returns:

- DDSXML_DestinationTransferTypeEnum The enum object if found else UNKNOWN_DATA_TYPE

2.2.25 DDSXML_DestinationStatesEnum Class Reference

This class is defines all destination states.

The Class diagram representing the DDSXML_ DestinationStatesEnum Class is provided in Figure 2.2.25.-1, DDSXML_ DestinationStatesEnum UML Diagram.

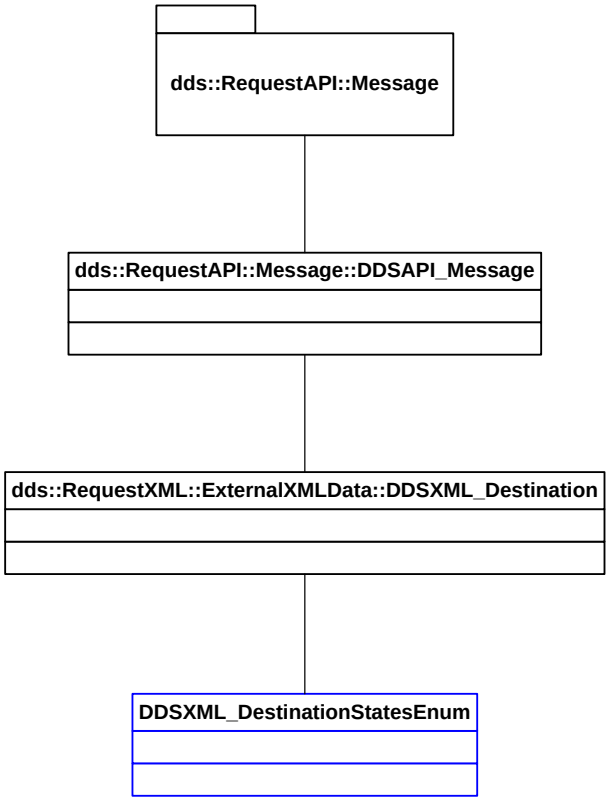

Figure 2.2.25-1, DDSXML_ DestinationStatesEnum Class UML Diagram

2.2.25.1 DDSXML_ DestinationStatesEnum Class Attributes

- static DDSXML_DestinationStatesEnum DESTINATION_UNKNOWN
- static DDSXML_DestinationStatesEnum DESTINATION_VALIDATED
- static DDSXML_DestinationStatesEnum DESTINATION_UNVALIDATED
- static DDSXML_DestinationStatesEnum DESTINATION_FAILED

2.2.25.2 DDSXML_ DestinationStatesEnum Class Enumerations

- static DDSXML_DestinationStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationStatesEnum.DESTINATION_UNKNOWN – Initial value: new DDSXML_DestinationStatesEnum("DESTINATION_UNKNOWN", index++)
- static DDSXML_DestinationStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationStatesEnum.DESTINATION_VALIDATED – Initial value: new DDSXML_DestinationStatesEnum("DESTINATION_VALIDATED", index++)
- static DDSXML_DestinationStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationStatesEnum.DESTINATION_UNVALIDATED – Initial value: new

DDSXML_DestinationStatesEnum("DESTINATION_UNVALIDATED", index++)

- static DDSXML_DestinationStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationStatesEnum.DESTINATION_FAILED – Initial value: new DDSXML_DestinationStatesEnum("DESTINATION_FAILED", index++)

2.2.25.3 DDSXML_ DestinationStatesEnum Class Functions

2.2.25.3.1 DDSXML_DestinationStatesEnum.findByName

static DDSXML_DestinationStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationStatesEnum.findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just returns DESTINATION_UNKNOWN

Parameters:

- name The string representing an enums name

Returns:

- DDSXML_DestinationStatesEnum The enum object if found else UNKNOWN_DATA_TYPE

2.2.25.4 DDSXML_DestinationStatesEnum.findByValue

static DDSXML_DestinationStatesEnum dds.RequestXML.ExternalXMLData.DDSXML_DestinationStatesEnum.findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just returns DESTINATION_UNKNOWN

Parameters:

- value The int representing an enums value

Returns:

- DDSXML_DestinationStatesEnum The enum object if found else UNKNOWN_DATA_TYPE

2.2.26 DDSXML_UserList Class Reference

This class is responsible for handling the User List XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API.

The Class diagram representing the DDSXML_ UserList Class is provided in Figure 2.2.26.-1, DDSXML_ UserList UML Diagram.

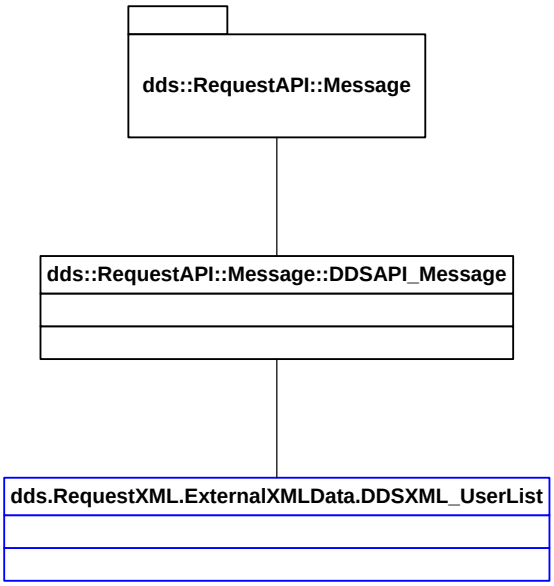

Figure 2.2.26-1, DDSXML_ UserList Class UML Diagram

2.2.26.1 DDSXML_ UserList Class Attributes

- static final String dds.RequestXML.ExternalXMLData.DDSXML_UserList.XML_USER_LIST_HEADER – Initial value: "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\" ?>" + <DDSXML_UserList><User></User></DDSXML_UserList>" – The XML header to create this message type.
- static final String dds.RequestXML.ExternalXMLData.DDSXML_UserList.XML_USER_TAG = "User" – The XML tag to create this XML type.
- static final String dds.RequestXML.ExternalXMLData.DDSXML_UserList.XML_USER_LIST_TAG = "UserList" – The XML tag to create this XML type.

2.2.26.2 DDSXML_ UserList Class Functions

2.2.26.2.1 DDSXML_UserList.equals

boolean dds.RequestXML.ExternalXMLData.DDSXML_UserList.equals (

DDSXML_UserList userList

)

Compare operator ==

Parameters:

- user The DDSXML_UserList that the data is to be compared to.

Returns:

- boolean true if the two are equal.

2.2.26.2.2 DDSXML_UserList.validate

boolean dds.RequestXML.ExternalXMLData.DDSXML_UserList.validate (

)

This method validates the data.

Returns:

- boolean true if the data is valid.

2.2.26.2.3 Deleted

2.2.26.2.4 DDSXML_UserList.getUsers

Vector dds.RequestXML.ExternalXMLData.DDSXML_UserList.getUsers (

)

Get the list of all Users which the requestor has access to as a vector.

Returns:

- Vector The vector that contains all users which the requestor has access to.. This vector may become invalid if a user is deleted from the list.

2.2.26.2.5 DDSXML_UserList.getNumberOfUsers

int dds.RequestXML.ExternalXMLData.DDSXML_UserList.getNumberOfUsers (

)

This method returns the number of users for this user

Returns:

- int The number of users for this user

2.2.26.2.6 DDSXML_UserList.findUser

boolean dds.RequestXML.ExternalXMLData.DDSXML_UserList.findUser (

DDSXML_User user

)

This method returns true if the user exists in the list.

Parameters:

- user The user to check for

Returns:

- boolean True if the user is found in the list.

2.2.26.2.7 DDSXML_UserList.findUser

DDSXML_User dds.RequestXML.ExternalXMLData.DDSXML_UserList.findUser (

String userName,

String userRole

)

This method returns true if the user exists in the list. The caller owns the user returned.

Parameters:

- userName The name of the user to look for.
- userRole The Role of the user to look for

Returns:

- DDSXML_User* The user or 0 if not found in the list.

2.2.26.2.8 DDSXML_UserList.validateFields

boolean dds.RequestXML.ExternalXMLData.DDSXML_UserList.validateFields (

boolean userListFlag

)

This method validates the XML.

Parameters:

- userListFlag If the flag is set validate this field

Returns:

- boolean true if the data is valid.

2.2.27 DDSXML_ClientTypesEnum Class Reference

This class is defines all command states This must match the C++.

The Class diagram representing the DDSXML_ ClientTypesEnum Class is provided in Figure 2.2.27.-1, DDSXML_ ClientTypesEnum UML Diagram.

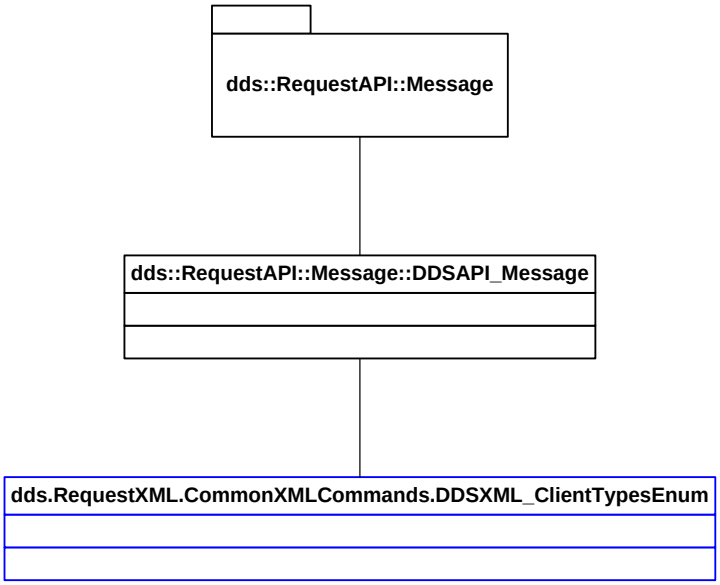

Figure 2.2.27-1, DDSXML_ ClientTypesEnum Class UML Diagram

2.2.27.1 DDSXML_ ClientTypesEnum Class Attributes

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.CLIENT_API = 0
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.GUI_API = 1
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.HANDLER_API = 2
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.FILE_TRANSFER_API = 3
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.SERVER_API = 4
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.UNKNOWN_API = 5
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.MIN_VALUE = CLIENT_API

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.MAX_VALUE = UNKNOWN_API + 1

2.2.27.2 DDSXML_ ClientTypesEnum Class Enumerations

- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.CLIENT_API_ENUM – Initial value: new DDSXML_ClientTypesEnum("CLIENT_API", CLIENT_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.GUI_API_ENUM – Initial value: new DDSXML_ClientTypesEnum("GUI_API", GUI_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.HANDLER_API_ENUM – Initial value: new DDSXML_ClientTypesEnum("HANDLER_API", HANDLER_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.FILE_TRANSFER_API_ENUM – Initial value: new DDSXML_ClientTypesEnum("FILE_TRANSFER_API", FILE_TRANSFER_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.SERVER_API_ENUM – Initial value: new DDSXML_ClientTypesEnum("SERVER_API", SERVER_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.UNKNOWN_API_ENUM – Initial value: new DDSXML_ClientTypesEnum("UNKNOWN_API", UNKNOWN_API)

2.2.27.3 DDSXML_ ClientTypesEnum Class Functions

2.2.27.3.1 DDSXML_ClientTypesEnum.findByName

static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just returns UNKNOWN_API

Parameters:

- val The string representing an enums name

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

2.2.27.3.2 DDSXML_ClientTypesEnum.findByValue

static DDSXML_ClientTypesEnum
dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.findByValue
(

int value

)

Find the enum that corresponds to the given string. If can't find it just returns UNKNOWN_CMD

Parameters:

- val The int representing an enums value

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

2.2.28 DDSXML_DataProductIDList Class Reference

This class is responsible for handling the data product List XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. This is used for the initial configuration from the Server. The request uses an ID list only.

The Class diagram representing the DDSXML_ DataProductIDList Class is provided in Figure 2.2.28.-1, DDSXML_ DataProductIDList UML Diagram.

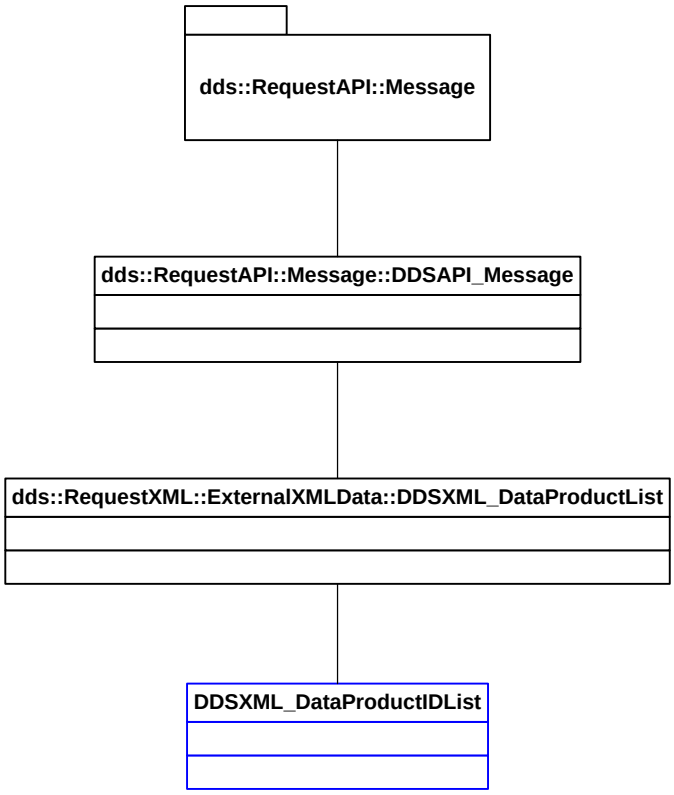

Figure 2.2.28-1, DDSXML_ DataProductIDList Class UML Diagram

2.2.28.1 DDSXML_ DataProductIDList Class Attributes

- static String XML_DATA_PRODUCT_TAG = "DataProductID"
- static String XML_DATA_PRODUCT_LIST_TAG = "DataProductIDList"

2.2.28.2 DDSXML_ DataProductIDList Class Functions

2.2.28.2.1 DDSXML_DataProductIDList::equals

boolean dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::equals (

DDSXML_DataProductIDList dataProductIDList

)

Compare equals

Parameters:

- dataProductIDList The DDSXML_DataProductIDList that the data is to be compared to.

Returns:

- boolean true if the two are equal.

2.2.28.2.2 DDSXML_DataProductIDList::validate

boolean dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::validate (

)

This method validates the data.

Returns:

- boolean true if the data is valid.

2.2.28.2.3 Deleted

2.2.28.2.4 DDSXML_DataProductIDList::addDataProduct

boolean dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::addDataProduct (

DDSXML_DataProductID dataProduct

)

This method adds the data Product to this list.

Parameters:

- dataProduct The data Product

Returns:

- boolean true if the product is added.

2.2.28.2.5 DDSXML_DataProductIDList::deleteDataProduct

boolean dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::deleteDataProduct (

String dataProductID

)

Allows the caller to delete a data product from the list.

Parameters:

- dataProductID Deletes data product based on data product ID.

Exceptions:

- DDSAPI_Exception if the dp ID is "", does not exist in the DDSXML_DataProductIDList or an error occurred during the manipulation of the DOM.

Returns:

- boolean true if the product is deleted.

2.2.28.2.6 DDSXML_DataProductIDList::getDataProduct

DDSXML_DataProductID
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::getDataProduct (

String dataProductID

)

This method returns the dataProduct identified by the dataProduct ID if it exists in the system

Parameters:

- dataProductID The dataProductID to check for

Returns:

- DDSXML_DataProductID The data product ID.

2.2.28.2.7 DDSXML_DataProductIDList::getDataProducts

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::getDataProducts (

)

This method returns the DataProducts.

Returns:

- Vector The Data Products

2.2.28.2.8 DDSXML_DataProductIDList::getDataProductIDs

Vector
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::getDataProductIDs (

)

This method returns the DataProducts IDs. Caller owns the data products returned.

Returns:

- Vector The Data Product's IDs

2.2.28.2.9 DDSXML_DataProductIDList::hasDataProductID

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::hasDataProductID (

String dataProductID

)

This method returns true if the dataProduct ID exists in the system

Parameters:

- dataProductID The dataProductID to check for

Returns:

- boolean True if ID is found.

2.2.28.2.10 DDSXML_DataProductIDList::hasDataProductID

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::hasDataProductID (

String dataProductID,

boolean sendStatus

)

This method returns true if the dataProduct ID exists in the system

Parameters:

- dataProductID The dataProductID to check for
- sendStatus True will send a status message to the User that is logged in.

Returns:

- boolean True if ID is found.

2.2.28.2.11 DDSXML_DataProductIDList::getNumberOfDataProducts

int
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::getNumberOfDataProducts (

)

This method returns the number of data products for this user

Returns:

- int The number of dataProducts for this user

2.2.28.2.12 DDSXML_DataProductIDList::validateFields

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DataProductIDList::validateFields (

)

This method validates the XML.

Returns:

- boolean true if the data is valid.

2.2.29 DDSXML_Longitude Class Reference

This class is responsible for handling the user XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. The DDS system uses a username and user role as one unique user. All searches are base on the user name and user role as a key.

The Class diagram representing the DDSXML_ Longitude Class is provided in Figure 2.2.29.-1, DDSXML_ Longitude UML Diagram.

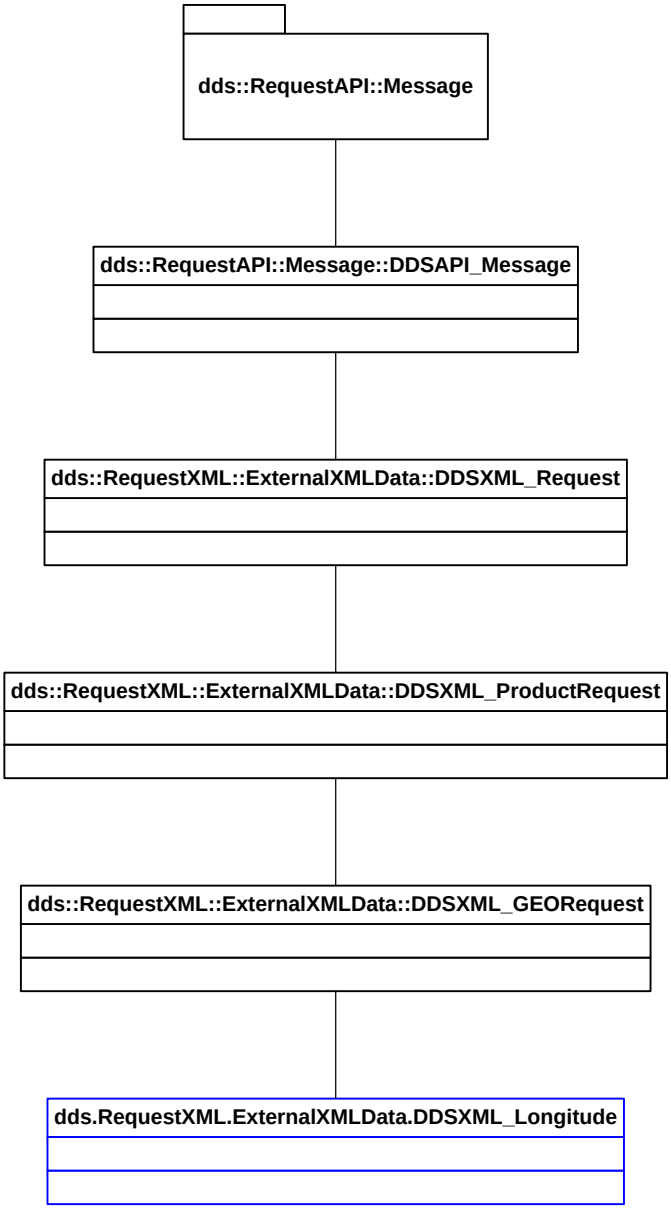

Figure 2.2.29-1, DDSXML_ Longitude Class UML Diagram

2.2.29.1 DDSXML_ Longitude Class Functions

2.2.29.1.1 DDSXML_Longitude.setLowerRightLongitude

static DDSXML_Longitude
dds.RequestXML.ExternalXMLData.DDSXML_Longitude.setLowerRightLongitude (

boolean isNegative,

int degrees,

int minutes,

float seconds

)

This method sets the Lower Right Longitude

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the longitude as an integer.
- minutes The minutes value for the longitude as an integer.
- seconds The seconds value for the longitude as a float. .

Returns:

- DDSXML_Longitude The Longitude or NULL if not valid.

2.2.29.2 DDSXML_Longitude.setUpperLeftLongitude

static DDSXML_Longitude
dds.RequestXML.ExternalXMLData.DDSXML_Longitude.setUpperLeftLongitude (

boolean isNegative,

int degrees,

int minutes,

float seconds

)

This method sets the Upper Left Longitude.

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the longitude as an integer.
- minutes The minutes value for the longitude as an integer.
- seconds The seconds value for the longitude as a float. .

Returns:

- DDSXML_Longitude The Longitude or NULL if not valid.

2.2.30 DDSXML_Latitude Class Reference

This class is responsible for handling the user XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. The DDS system uses a username and user role as one unique user. All searches are based on the user name and user role as a key.

The Class diagram representing the DDSXML_ Latitude Class is provided in Figure 2.2.30.-1, DDSXML_ Latitude UML Diagram.

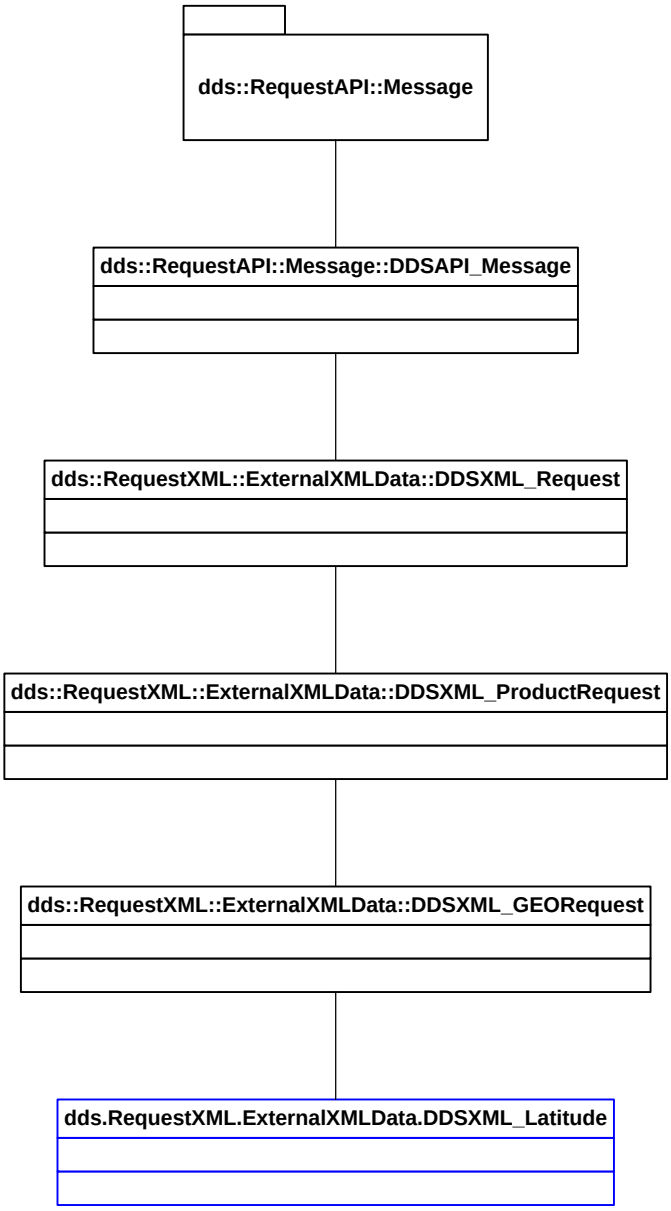

Figure 2.2.30-1, DDSXML_ Latitude Class UML Diagram

2.2.30.1 DDSXML_ Latitude Class Functions

2.2.30.1.1 DDSXML_Latitude.setLowerRightLatitude

```
static DDSXML_Latitude
dds.RequestXML.ExternalXMLData.DDSXML_Latitude.setLowerRightLatitude (
    boolean isNegative,
    int degrees,
    int minutes,
    float seconds
  )
```

This method sets the Lower Right Latitude.

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the latitude as an integer.
- minutes The minutes value for the latitude as an integer.
- seconds The seconds value for the latitude as a float.

Returns:

- DDSXML_Latitude The Latitude or NULL if not valid.

2.2.30.1.2 DDSXML_Latitude.setUpperLeftLatitude

```
static DDSXML_Latitude
dds.RequestXML.ExternalXMLData.DDSXML_Latitude.setUpperLeftLatitude (
    boolean isNegative,
    int degrees,
    int minutes,
    float seconds
  )
```

This method sets the Upper Left Latitude.

Parameters:

- isNegative Specifies if the LatLong is negative
- degrees The degrees value for the latitude as an integer.
- minutes The minutes value for the latitude as an integer.
- seconds The seconds value for the latitude as a float.

Returns:

- DDSXML_Latitude The Latitude or NULL if not valid.

2.2.31 DDSXML_ResultSet Class Reference

This class provides a wrapper for the DDSAPI_Granule list. It provides convenience classes for accessing the DDSAPI_Granule list.

The Class diagram representing the DDSXML_ ResultSet Class is provided in Figure 2.2.31.-1, DDSXML_ ResultSet UML Diagram.

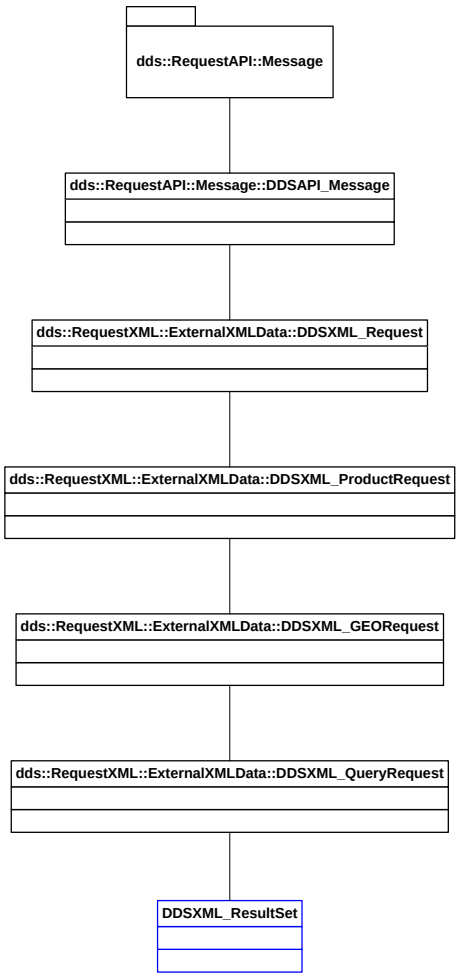

Figure 2.2.31-1, DDSXML_ ResultSet Class UML Diagram

2.2.31.1 DDSXML_ ResultSet Class Functions

2.2.31.1.1 DDSXML_ResultSet.validate

boolean dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.validate (

)

This method validates the data.

Returns:

- boolean true if the data is valid.

2.2.31.1.2 Deleted

2.2.31.1.3 DDSXML_ResultSet.getNumberReturned

int dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.getNumberReturned (

)

This method returns the number of GranuleMetaData objects returned. This number may differ from the number found in the query if the results set is greater than the max threshold. If the value is 0 then an error has occurred in processing this query or there were none found.

Returns:

- int Number of GranuleMetaDatas in the List

2.2.31.1.4 DDSXML_ResultSet.addGranuleMetaData

boolean dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.addGranuleMetaData (

DDSXML_GranuleMetaData granuleMetaData,

String granuleMetaDataName

)

This method adds a GranuleMetaData.

Parameters:

- GranuleMetaData - The GranuleMetaData to add

Returns:

- boolean true if the data is valid.

Exceptions:

- RequestException

2.2.31.1.5 DDSXML_ResultSet.getGranuleMetaDataIDs

Vector dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.getGranuleMetaDataIDs (

)

This method returns a vector of GranuleMetaData ID.

Returns:

- Vector of GranuleMetaData IDs as String

2.2.31.1.6 DDSXML_ResultSet.getGranuleMetaData

DDSXML_GranuleMetaData
dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.getGranuleMetaData (

String granuleMetaDataID

)

This method returns a a DDSXML_GranuleMetaData.

Parameters:

- GranuleMetaDataId - The ID of the GranuleMetaData

Returns:

- DDSXML_GranuleMetaData The granule metadata

2.2.31.1.7 DDSXML_ResultSet.getGranuleMetaDatas

Vector dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.getGranuleMetaDatas (

)

This method returns a vector of GranuleMetaDatas

Returns:

- Vector of GranuleMetaDatas

2.2.31.1.8 DDSXML_ResultSet.validateFields

boolean dds.RequestXML.ExternalXMLData.DDSXML_ResultSet.validateFields (

boolean resultSetFlag

)

This method validates the XML.

Parameters:

- resultSetFlag If the flag is set validate this field

Returns:

- boolean true if the data is valid.

2.2.32 DDSXML_ResultSetList Class Reference

This class is responsible for handling the resultSet list XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API.

The Class diagram representing the DDSXML_ ResultSetList Class is provided in Figure 2.2.32.-1, DDSXML_ ResultSetList UML Diagram.

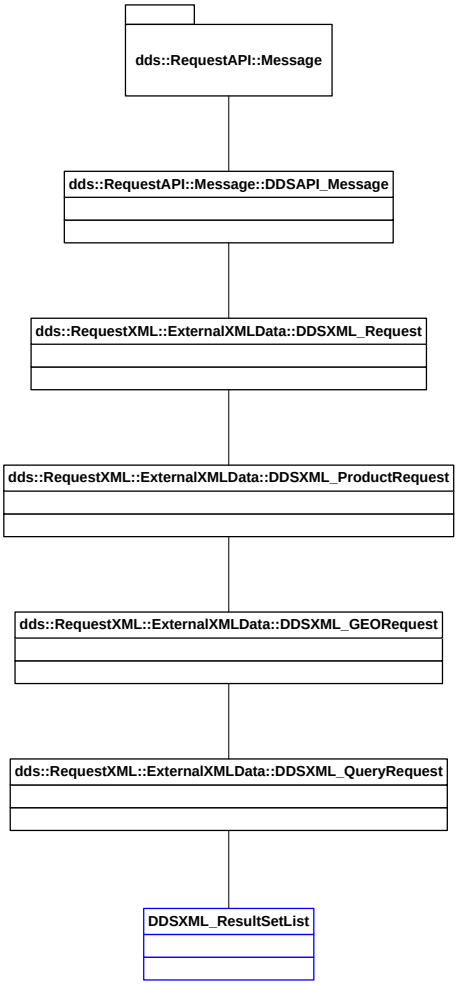

Figure 2.2.32-1, DDSXML_ ResultSetList Class UML Diagram

2.2.32.1 DDSXML_ ResultSetList Class Attributes

- static String dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.XML_RESULT_SET_TAG = "ResultSetLists" – The XML header to create this message type.
- static String dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.XML_RESULT_TAG = "ResultSetTag" – The XML tag to create this XML type.
- protected HashMap dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.myResultSetList = new HashMap() – The map of resultSets we have. This is indexed by resultSet key for faster lookup of resultSets.

2.2.32.2 DDSXML_ ResultSetList Class Functions

2.2.32.2.1 DDSXML_ResultSetList.validate

boolean dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.validate (

)

This method validates the data.

Returns:

- boolean true if the data is valid.

2.2.32.2.2 Deleted

2.2.32.2.3 DDSXML_ResultSetList.getResultSets

Vector dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.getResultSets (

)

Get the list of all resultSets as a vector. Caller owns the list returned.

Returns:

- Vector The vector that contains all resultSets. This resultSet may become invalid if a resultSet is deleted from the list.

2.2.32.2.4 DDSXML_ResultSetList.getNumberOfResultSets

int
dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.getNumberOfResultSets (

)

Get the list of all resultSets as a vector.

Returns:

- int Number of resultSets

2.2.32.2.5 DDSXML_ResultSetList.addResultSet

boolean dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.addResultSet (

DDSXML_ResultSet resultSet

)

Add the resultSet

Parameters:

- resultSet Result set to add

Returns:

- boolean True if the result set is added OK.

2.2.32.2.6 DDSXML_ResultSetList.validateFields

boolean dds.RequestXML.ExternalXMLData.DDSXML_ResultSetList.validateFields (

boolean resultSetListFlag

)

This method validates the XML.

Parameters:

- resultSetListFlag If the flag is set validate this field

Returns:

- boolean true if the data is valid.

2.2.33 DDSXML_DestinationEntry Class Reference

This class is responsible for handling the destination Entry XML.

It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. Some of the method names are left to be compatible with the old API. This old class was(is) exposed to external users so we can not change the method names now.

2.2.33.1 DDSXML_DestinationEntry Class Functions

2.2.33.1.1 DDSXML_DestinationEntry::getDestinationEntryName

String

dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getDestinationEntryName (

)

This method retrieves the Destination Entry Name as a string.

Returns:

- String The destination Name or "" if not valid

2.2.33.1.2 DDSXML_DestinationEntry::getFTPUserPassword

String

dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getFTPUserPassword (

)

This method retrieves the FTP Password attribute's value as a string. The value returned will be cleartext.

Returns:

- String The FTP password as a string or "" if not valid.

2.2.33.1.3 DDSXML_DestinationEntry::getPath

String dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getPath (

)

This method retrieves the myPath attribute's value as a string.

Returns:

- String The destination path as a string or "" if not valid.

2.2.33.1.4 DDSXML_DestinationEntry::getHostName

String
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getHostName (

)

This method returns the destination host name (or IP) Note that this is really a valid hostname or IP address that can be used for FTP.

Returns:

- String The Host name as a string or "" if not valid.

2.2.33.1.5 DDSXML_DestinationEntry::getFTPUserName

String
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getFTPUserName (

)

This method retrieves the FTP Username attribute's value as a string.

Returns:

- String The FTP username associated with this destination or "" if not valid..

2.2.33.1.6 DDSXML_DestinationEntry::getOwner

DDSXML_User
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getOwner (

)

This method retrieves the owner of this destinationEntry.

Returns:

- DDSXML_User The owner of this destination or NULL if not valid.

2.2.33.1.7 DDSXML_DestinationEntry::getState

DDSXML_DestinationStatesEnum
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getState (

)

This method retrieves the last state of this destinationEntry.

Returns:

- DDSXML_DestinationStatesEnum The state associated with this destinationEntry

2.2.33.1.8 DDSXML_DestinationEntry::getTransferType

DDSXML_DestinationTransferTypeEnum
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getTransferType (

)

This method retrieves the transfer Type of this destination

Returns:

- DDSXML_DestinationTransferTypeEnum The transfer Type associated with this destination

2.2.33.1.9 DDSXML_DestinationEntry::setDestinationEntryName

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setDestinationEntryName (

String destinationEntryName

)

This method sets the name of the destination Entry. This is a user defined name.

Parameters:

- destinationEntryName The name used to set destination name.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.10 DDSXML_DestinationEntry::setFTPUserPassword

boolean dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setFTPUserPassword (

String password

)

This method sets the FTP Password.

Parameters:

- password The FTP password being used to set myPassword.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.11 DDSXML_Destination::setPath

boolean dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setPath (

String Path

)

This method sets the Path.

Parameters:

- path The path being used to set myPath.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.12 DDSXML_DestinationEntry::setHostName

boolean

dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setHostName (

String hostName

)

This method sets the destination host name Note that this is really a valid hostname or IP address that can be used for FTP.

Parameters:

- hostName The host name.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.13 DDSXML_DestinationEntry::setFTPUserName

boolean

dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setFTPUserName (

String userName

)

This method sets the FTP user.

Parameters:

- userName The user name.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.14 DDSXML_DestinationEntry::setOwner

boolean dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setOwner (

DDSXML_User owner

)

This method sets the owner of this destination.

Parameters:

- owner The owner of this destination.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.15 DDSXML_DestinationEntry::setState

boolean dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setState (

DDSXML_DestinationStatesEnum state

)

This method sets the state attribute in the Destination class.

Parameters:

- state The state of the Destination.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.16 DDSXML_DestinationEntry::setTransferType

boolean
dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setTransferType (

DDSXML_DestinationTransferTypeEnum transferType

)

This method sets the transfer Type attribute in the Destination class. This will allow the destinationEntry to be sent using different transfer methods.

Parameters:

- transferType The transfer Type of the Destination.

Returns:

- boolean True if set OK, False if not.

2.2.33.1.17 DDSXML_DestinationEntry::getUserIndex

String

dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::getUserIndex (

)

This method returns the user Destination index.

Returns:

- String The user Index as a string or "" if not valid.

2.2.33.1.18 DDSXML_DestinationEntry::setFTPUser

boolean

dds::RequestXML::ExternalXMLData::DDSXML_DestinationEntry::setFTPUser (

String userName,

String userPassword

)

This method sets the FTP user.

Parameters:

- username The user name used to set myUsername.
- userPassword The password being used to set myPassword.

Returns:

- boolean True if set OK, False if not.

3. JAVA JMS DOCUMENTATION

The IDPS Java JMS interface allows users to interface with the IDPS/DDS via a messaging service. The javax JAR file is necessary for doing JMS messaging: javax.jms.*, javax.naming.*, and java.util.Properties.

In order to interact with IDPS/DDS via a JMS service, there are two components, the IDPS JMS Server executable (JMSServer) and the IDPS JMS Client API libraries (JMSClient). The JMSServer facilitates message communications between a JMS service and IDPS/DDS. The JMS Client libraries are employed by the user to format and parse DDSXML messages sent or received over the JMS interface.. See Figure 3.0-1, IDPS JMS Interface Graphical Depiction, for a graphical depiction of the intended implementation. Messages sent to the JMS service receive responses (two-way communication).

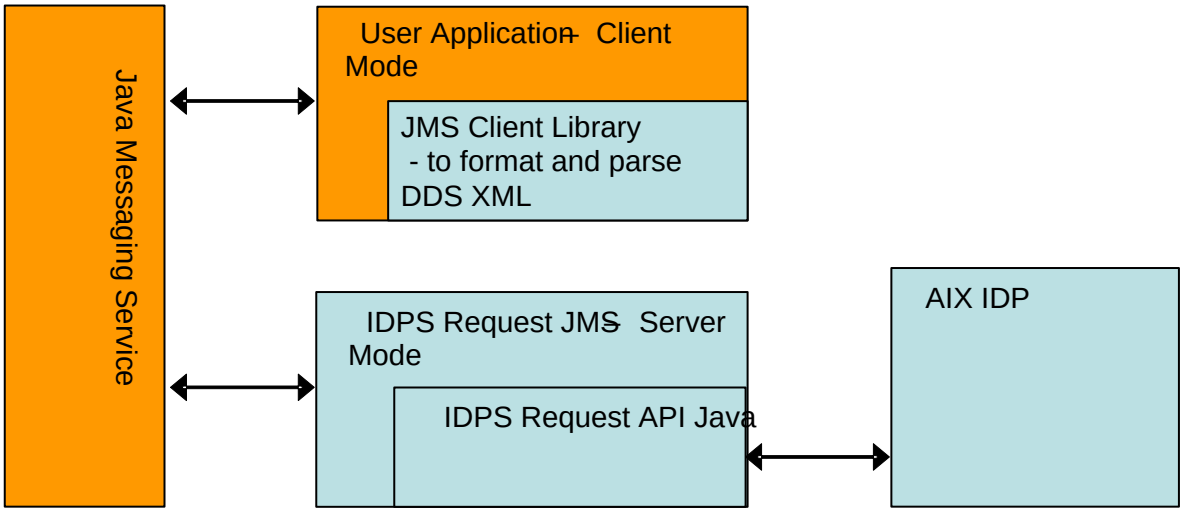

Figure 3.0-1, IDPS JMS Interface Graphical Depiction

In order to make use of the IDPS JAVA JMS interface, the environment variables listed in Section 3.2.1, Environment Variables, must be set up and the JMS classes relating to the establishment of sessions, subsessions, and topics as well as those relating to the publishing of and subscribing to messages must be instantiated. These allow the user application to communicate with the JMS service and connect to the IDPS JMS topics.

The user application needs to implement the MessageListener interface and the onMessage method in order to receive replies from the DDS JMS Server. The user application uses the DDSXML classes provided in the JMS Client Library to parse the XML in the JMS text messages received from the DDS JMS Server application..

When creating a message, the client must provide the topic to reply to in the “Reply To” portion of the message. A temporary topic works well for this, but a Topic created by the JMS Administrator will work as well.

The IDPS/DDS System messages are a one-way message communication from the IDPS/DDS to a User Topic and do not require a command to be sent in order to receive. Initial login and subscription to this service must be made before message receipt is possible. The System messages are free-form messages that provide broadcasted messages from IDPS/DDS and do not relate necessarily to any specific user action. They may change and should not be considered reliable or timely.

3.1 Coding Conventions

The JAVA JMS API is written in Java and follows the Java Coding Convention as documented in Section 2.1, Java Coding Conventions, above.

3.2 Java JMS Module Documentation List

The IDPS Java JMS Client API provides a programmatic interface to XML based request messages. When the request specifications have been defined, the user application extracts the XML-based messages (via the DDSXML_Command.getXML() accessor) from the JMS Client interface and forwards them to a JMS service. The messages allow a user application to login, create a request, process a request, and perform other basic manipulations on their requests. As XML-based messages are received from the JMS service, the user application uses the DDSXML classes from the JMS Client library(via the DDSXML_Command constructor) to access the message details.. The class definitions that provide the message interface (message types, accessors, mutators) are in the dds.RequestXML package.

3.2.1 Environment Variables

Following is a list of the environment variables that need to be specified for use by the JMS Client.

- The **DDS_PORT** environment variable tells the JMS API application what port to use when communicating with the DDS Server.[2]
- The **DDS_HOST** environment variable tells the JMS API application the IP address of the machine that is hosting the DDS Server.[2]
- The **DDS_TIMEOUT** variable tells the JMS API how many minutes of inactivity before closing the session.[2]
- The **DDS_ROOT** variable identifies the location where IDPS log files should be stored if necessary.[2]
- The **WL_HOST** environment variable is used by the Client to identify the IP address of the machine hosting the BEA Weblogic server.[2,3]
- The **WL_PORT** environment variable is used by the Client to identify the port used to communicate with the Weblogic server.[2,3]
- The **WL_USERNAME** environment variable is used by the Client to hold the username for connecting to the Weblogic server.[2,3]
- The **WL_PASSWORD** environment variable is used by the Client to hold the password for connecting to the Weblogic server.[2,3]

- The **WL_TOPIC** environment variable is used by the Client to hold the name of the JMS Topic that is used for sending messages to the JMS API. It must match the Topic name defined in the Weblogic server.[2,3]
- The **DSTATICDATA** variable tells IDPS JMS Server the location of configuration files provided.[1,2]
- The **BEA_HOME** variable is used to identify the location of the BEA Weblogic Commercial-Off-The-Shelf (COTS) JAR file (weblogic.jar). This JAR file is necessary to connect to the Weblogic server.[1,2,3]
- The **JMS_JAVA_HOME** variable is used to identify the location the JMS COTS JAR files (jms.jar and javax.jms.jar). These JAR file are necessary to do JMS messaging.[2,3]
- The **JMS_TIMEOUT** variable tells the JMS API how many minutes of inactivity before Logging out the client. This is typically 30 minutes.[2]
- The **IDLE_PERIODS_UNTIL_TIMEOUT** variable tells the Weblogic server how many idle periods by the JMS Client before closing the connection to the JMS Client. This must be set on the JMS Client machine. [2,3]
- The **VBROKERDIR** variable identifies the location of Visibroker COTS JAR files (lib\vbjorb.jar, lib\vbjdev.jar, lib\lm.jar, lib\vbsec.jar).[2]
- The **XERCES_JAVA_HOME** variable identifies the location of Xerces COTS JAR files (xercesImpl.jar, xmlParserAPIs.jar).[2]
- The **DDSCAT1ROOT** variable identifies the location of IDPS DDS JAR files (lib\DDSRequestXML.jar, lib\DDSRequestAPI.jar, lib\JMSAPI.jar).[1,2]
- The **INFCAT1ROOT** variable identifies the location of IDPS INF JAR files (lib\InfUtilTim.jar, lib\InfCmnExc.jar, lib\InfInfoDist.jar, lib\InfUtilSec.jar, lib\InfUtilCfg.jar, lib\InfCmnXML.jar).[1,2]

Notes:

1. The IDPS Windows Installshield creates this environment variable when it's installed
2. Needed to run the JMS Server
3. Needed to run the JMS Client

3.3 Java JMS API Module Documentation List

The Java JMS Client library is a collection of classes that allow the users JMS client application to format and parse DDSXML messages sent or received through JMS Topics. The users JMS client application will use DDSXML messages to communicate, via the DDS JMS Server application, with the DDS Request Server. These messages allow the user to logon, create a request, process a request, and perform other functions available through the Java API.

3.3.1 DDSXML_Login Class Reference

This class is responsible for handling the Login command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a login command

The Class diagram representing the DDSXML_Login Class is provided in Figure 3.3.1-1, DDSXML_Login Class UML Diagram.

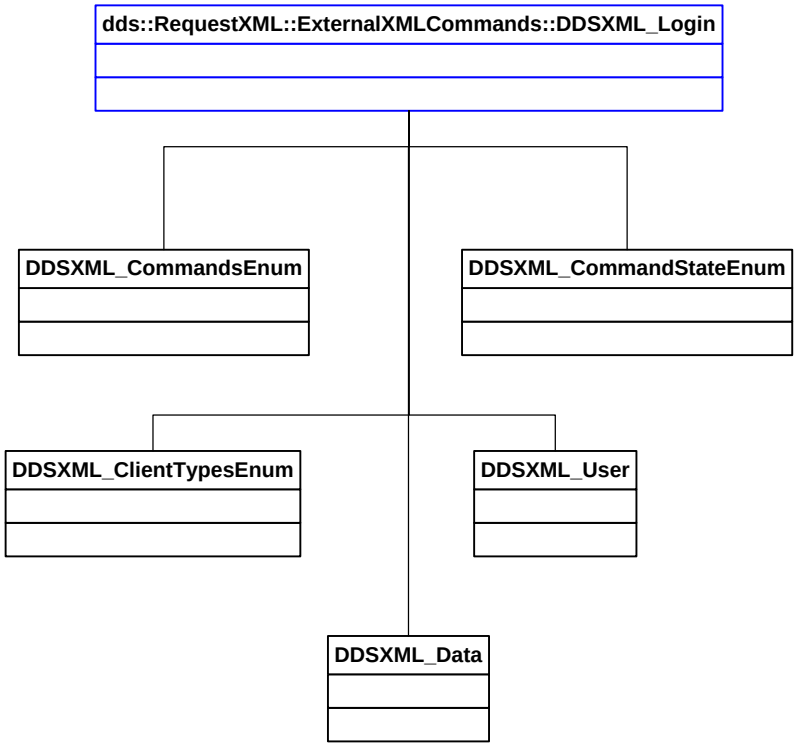

Figure 3.3.1-1, DDSXML_Login Class UML Diagram

3.3.1.1 DDSXML_Login Class Functions

3.3.1.1.1 DDSXML_Login::DDSXML_Login

```
dds::RequestXML::ExternalXMLCommands::DDSXML_Login::DDSXML_Login (
    String connectionID,
    String commandID,
    DDSXML_CommandsEnum command,
    DDSXML_CommandStateEnum commandState,
    String commandData,
    DDSXML_ClientTypesEnum clientType,
    DDSXML_User user,
    String dataID
  )
```

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.
- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text from a DDSXML_DataID that is the encrypted form of a password.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.1.1.2 DDSXML_Login::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_Login::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_User

Returns:

- The DDSXML_User object or null if none.

3.3.2 DDSXML_LoginResponse Class Reference

This class is responsible for handling the Login Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class contains the Request Server response to a user login command.

The Class diagram representing the DDSXML_LoginResponse Class is provided in Figure 3.3.2-1, DDSXML_LoginResponse Class UML Diagram.

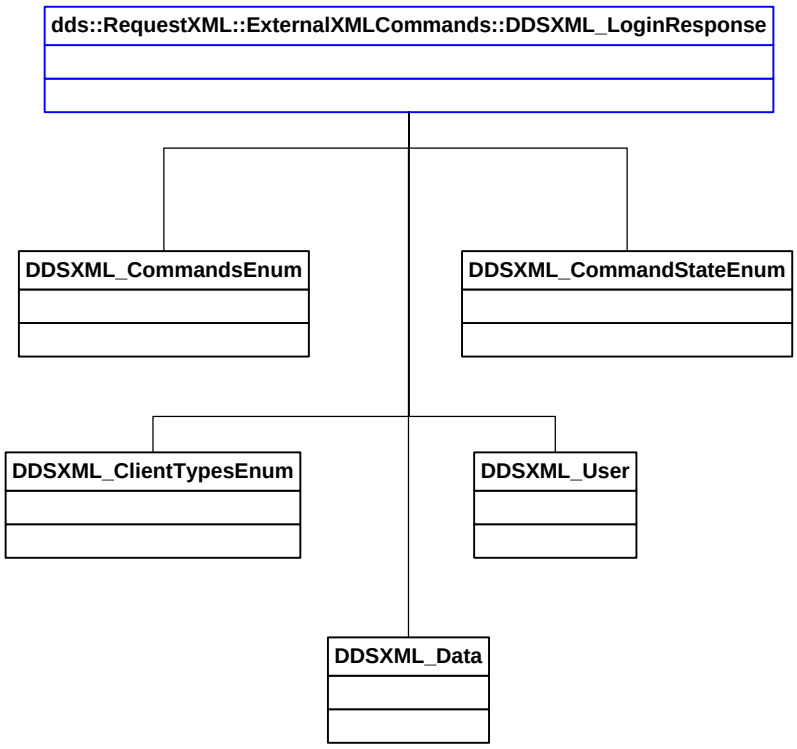

Figure 3.3.2-1, DDSXML_LoginResponse Class UML Diagram

3.3.2.1 DDSXML_LoginResponse Class Functions

3.3.2.1.1 DDSXML_LoginResponse::DDSXML_LoginResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_LoginResponse::DDSXML_LoginResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.
- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- Command The command to execute on the Server
- clientType The Type of client for this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.

3.3.2.1.2 DDSXML_LoginResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_LoginResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_NoData

Returns:

- The DDSXML_NoData object or null if none.

3.3.3 DDSXML_Logout Class Reference

This class is responsible for handling the Logout command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a logout command.

The Class diagram representing the DDSXML_Logout Class is provided in Figure 3.3.3-1, DDSXML_Logout Class UML Diagram.

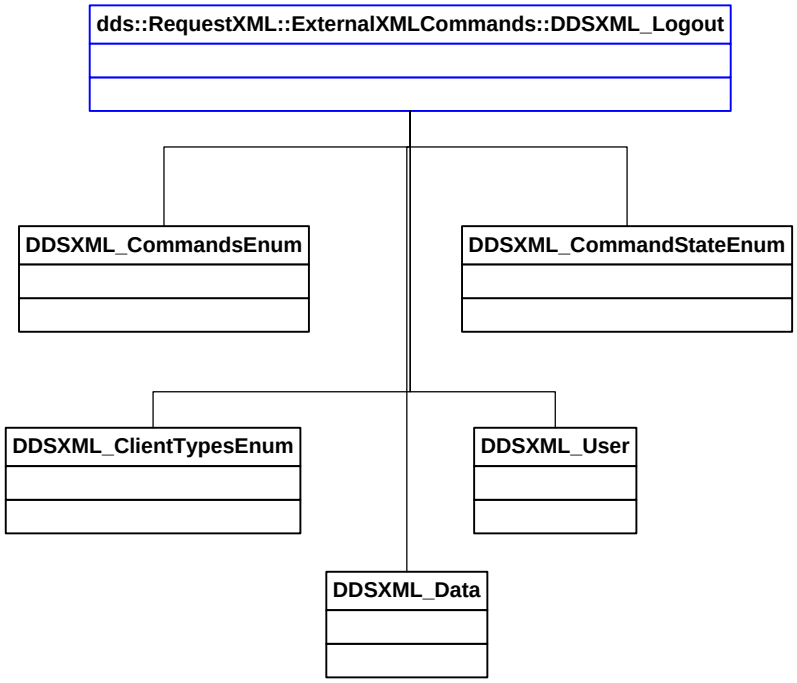

Figure 3.3.3-1, DDSXML_Logout Class UML Diagram

3.3.3.1 DDSXML_Logout Class Functions

3.3.3.1.1 DDSXML_Logout::DDSXML_Logout

dds::RequestXML::ExternalXMLCommands::DDSXML_Logout::DDSXML_Logout (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- Command The command to execute on the Server
- clientType The Type of client for this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.

3.3.3.1.2 DDSXML_Logout::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_Logout::getCommandDataFromXML (

)

This method retrieves the data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_User

Returns:

- The DDSXML_User object or null if none.

3.3.4 DDSXML_LogoutResponse Class Reference

This class is responsible for handling the Logout Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class contains the Request Server response to a user logout command

The Class diagram representing the DDSXML_LogoutResponse Class is provided in Figure 3.3.4-1, DDSXML_LogoutResponse Class UML Diagram.

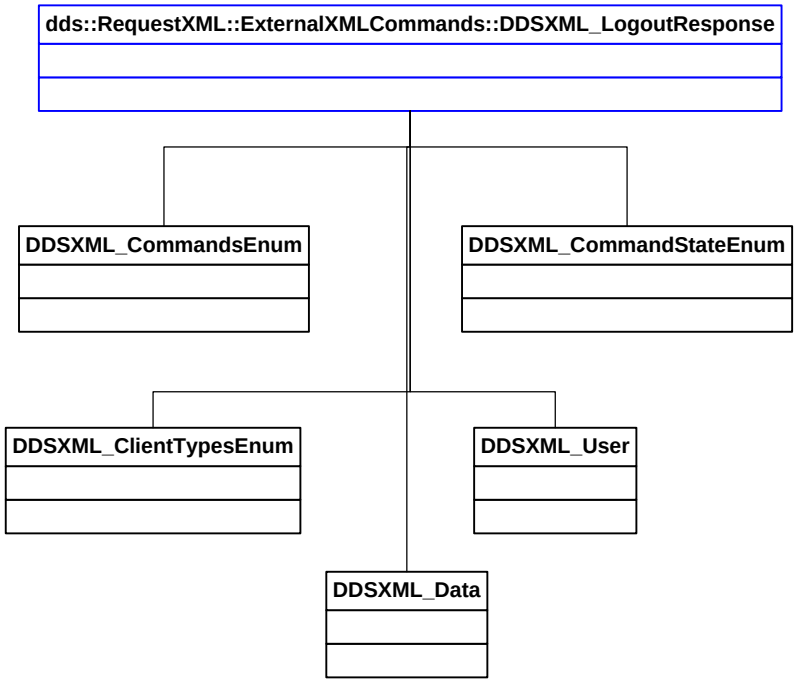

Figure 3.3.4-1, DDSXML_LogoutResponse Class UML Diagram

3.3.4.1 DDSXML_LogoutResponse Class Functions

3.3.4.1.1 DDSXML_LogoutResponse::DDSXML_LogoutResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_LogoutResponse::DDSXML_LogoutResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- DDSXML_LogoutResponse::DDSXML_LogoutResponse connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- Command The command to execute on the Server
- clientType The Type of client for this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.

3.3.4.1.2 DDSXML_LogoutResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_LogoutResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_NoData

Returns:

- The DDSXML_NoData object or null if none.

3.3.5 DDSXML_CreateRequest Class Reference

This class is responsible for handling the Create Request command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a create request command

The Class diagram representing the DDSXML_CreateRequest Class is provided in Figure 3.3.5-1, DDSXML_CreateRequest Class UML Diagram.

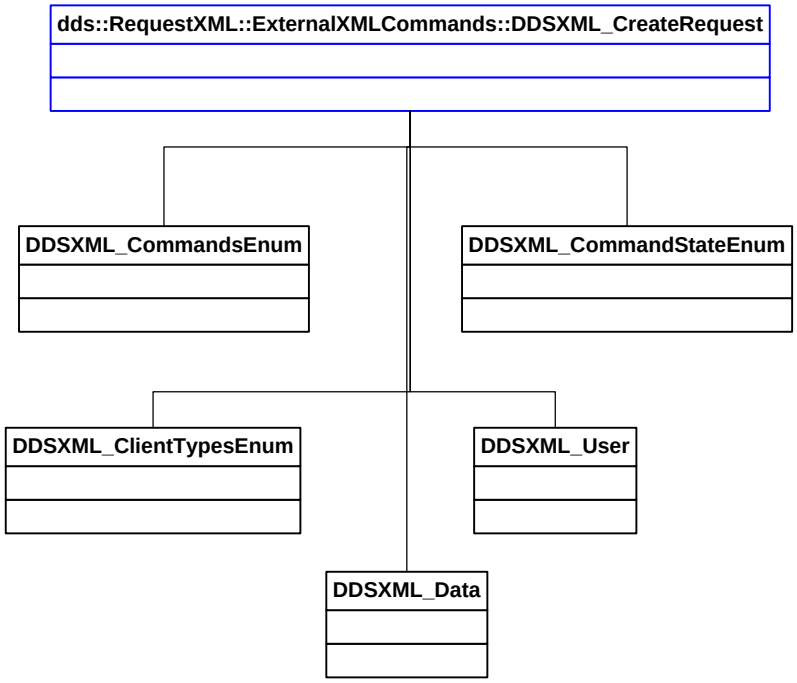

Figure 3.3.5-1, DDSXML_CreateRequest Class UML Diagram

3.3.5.1 DDSXML_CreateRequestClass Functions

3.3.5.1.1 DDSXML_CreateRequest::DDSXML_CreateRequest

dds::RequestXML::ExternalXMLCommands::DDSXML_CreateRequest::DDSXML_CreateRequest (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.5.1.2 DDSXML_CreateRequest::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_CreateRequest::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.6 DDSXML_CreateRequestResponse Class Reference

This class is responsible for handling the Create Request Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class enables a user to build a create request response command

The Class diagram representing the DDSXML_ CreateRequestResponse Class is provided in Figure 3.3.6-1, DDSXML_ CreateRequestResponse Class UML Diagram.

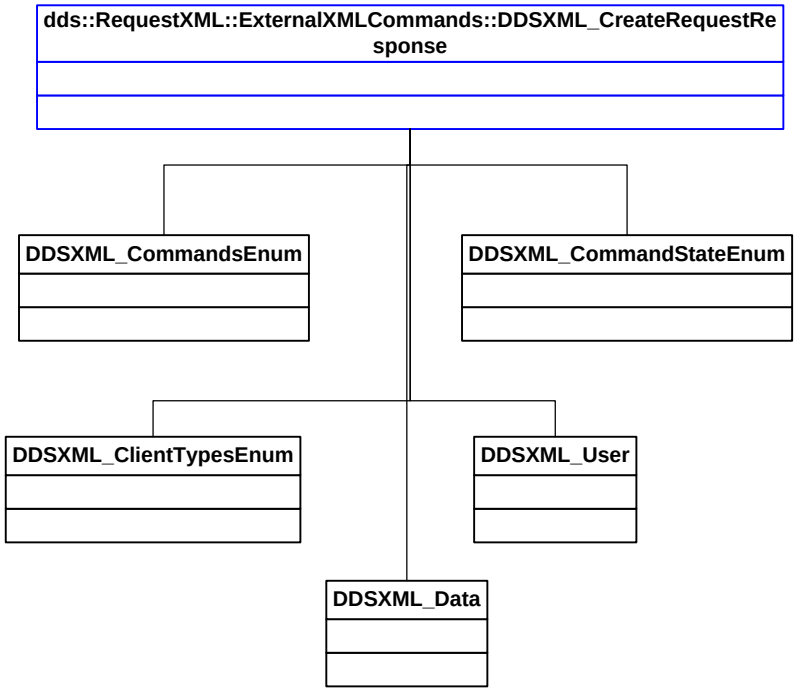

Figure 3.3.6-1, DDSXML_ CreateRequestResponse Class UML Diagram

3.3.6.1 DDSXML_ CreateRequestResponse Functions

3.3.6.1.1 DDSXML_ CreateRequestResponse::DDSXML_ CreateRequestResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_CreateRequestResponse::DDSXML_CreateRequestResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user, String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.
- commandID A unique ID for each command sent to the Server. This ID will be used in the response command

- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.6.1.2 DDSXML_CreateRequestResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_CreateRequestResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.7 DDSXML_ModifyRequest Class Reference

This class is responsible for handling the Modify Request command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a modify request command

The Class diagram representing the DDSXML_ModifyRequest Class is provided in Figure 3.3.7-1, DDSXML_ModifyRequest Class UML Diagram.

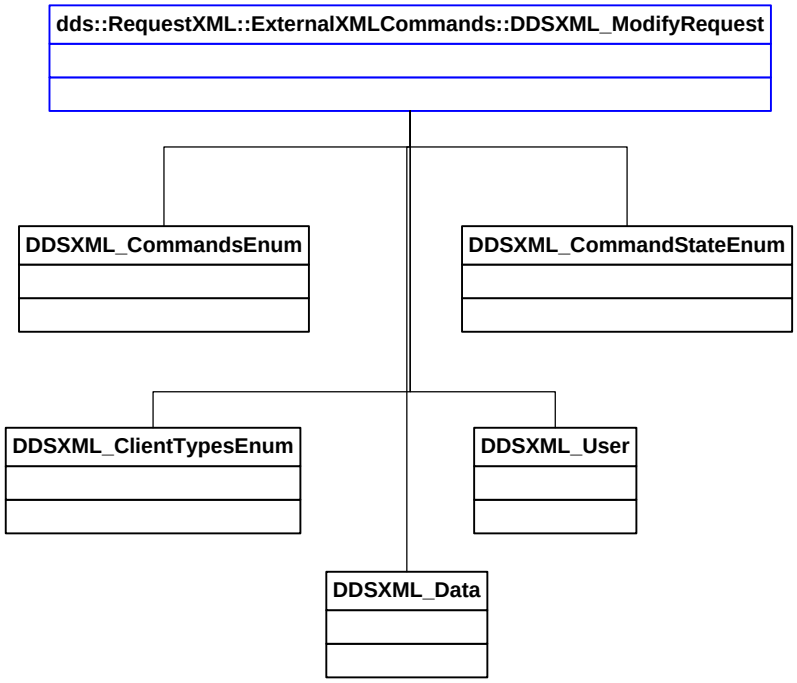

Figure 3.3.7-1, DDSXML_ModifyRequest Class UML Diagram

3.3.7.1 DDSXML_ModifyRequest Functions

3.3.7.1.1 DDSXML_ModifyRequest::DDSXML_ModifyRequest

dds::RequestXML::ExternalXMLCommands::DDSXML_ModifyRequest::DDSXML_ModifyRequest (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.7.1.2 DDSXML_ModifyRequest::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_ModifyRequest::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.8 DDSXML_ModifyRequestResponse Class Reference

This class is responsible for handling the Modify Request Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class enables the user to build a modify request response command

The Class diagram representing the DDSXML_ModifyRequestResponse Class is provided in Figure 3.3.8-1, DDSXML_ModifyRequestResponse Class UML Diagram.

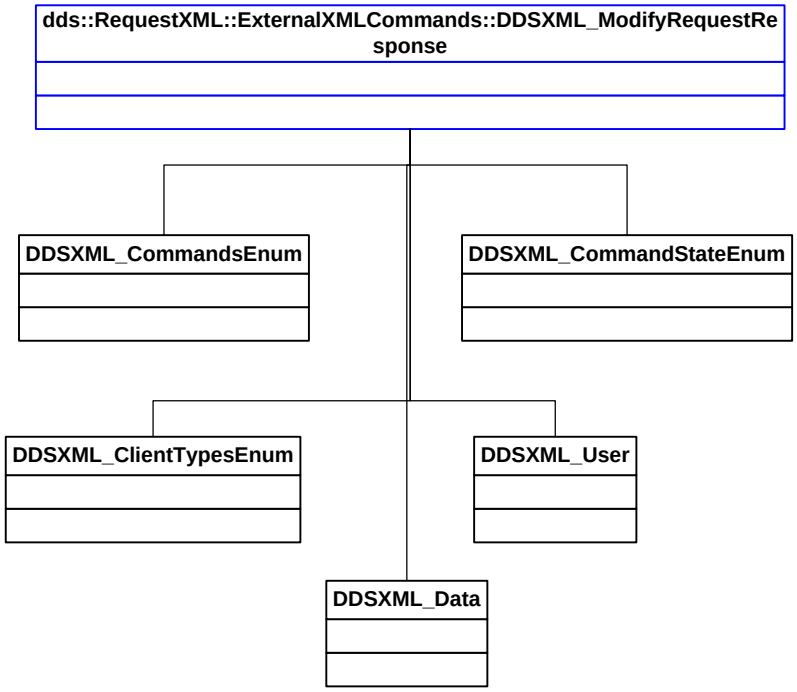

Figure 3.3.8-1, DDSXML_ModifyRequestResponse Class UML Diagram

3.3.8.1 DDSXML_ModifyRequestResponse Functions

3.3.8.1.1 DDSXML_ModifyRequestResponse::DDSXML_ModifyRequestResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_ModifyRequestResponse::DDSXML_ModifyRequestResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.8.1.2 DDSXML_ModifyRequestResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_ModifyRequestResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.9 DDSXML_DeleteRequest Class Reference

This class is responsible for handling the Delete Request command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a delete request command

The Class diagram representing the DDSXML_DeleteRequest Class is provided in Figure 3.3.9-1, DDSXML_DeleteRequest Class UML Diagram.

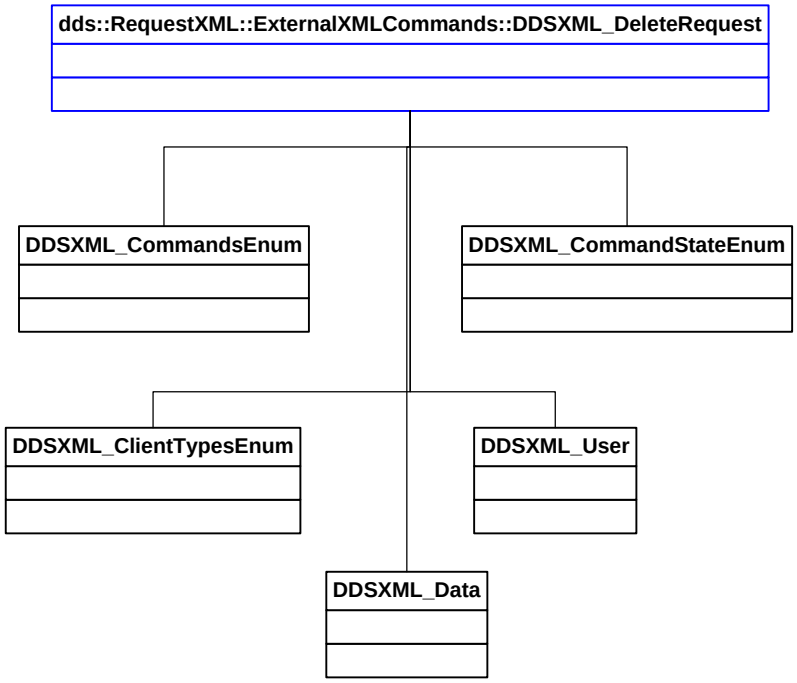

Figure 3.3.9-1, DDSXML_DeleteRequest Class UML Diagram

3.3.9.1 DDSXML_DeleteRequest Functions

3.3.9.1.1 DDSXML_DeleteRequest::DDSXML_DeleteRequest

dds::RequestXML::ExternalXMLCommands::DDSXML_DeleteRequest::DDSXML_DeleteRequest (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- Command The command to execute on the Server
- clientType The Type of client for this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.

3.3.9.1.2 DDSXML_DeleteRequest::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_DeleteRequest::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.10 DDSXML_DeleteRequestResponse Class Reference

This class is responsible for handling the Delete Request Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a delete request response command

The Class diagram representing the DDSXML_DeleteRequestResponse Class is provided in Figure 3.3.10-1, DDSXML_DeleteRequestResponse Class UML Diagram.

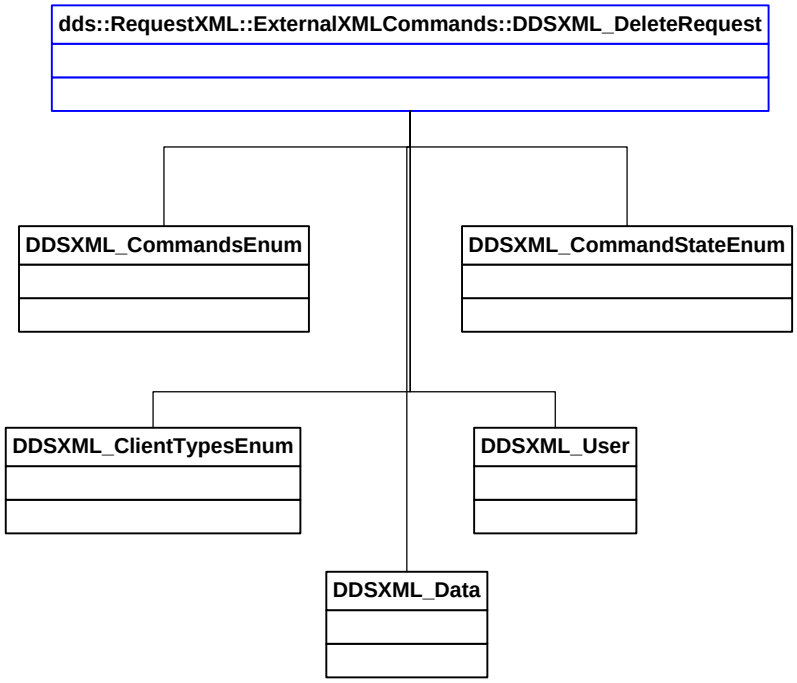

Figure 3.3.10-1, DDSXML_DeleteRequestResponse Class UML Diagram

3.3.10.1 DDSXML_DeleteRequestResponse Functions

3.3.10.1.1 DDSXML_DeleteRequestResponse::DDSXML_DeleteRequestResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_DeleteRequestResponse::DDSXML_DeleteRequestResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.10.1.2 DDSXML_DeleteRequestResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_DeleteRequestResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.11 DDSXML_SuspendRequest Class Reference

This class is responsible for handling the Suspend Request command XML.

It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a suspend request command

The Class diagram representing the DDSXML_SuspendRequest Class is provided in Figure 3.3.11-1, DDSXML_SuspendRequest Class UML Diagram.

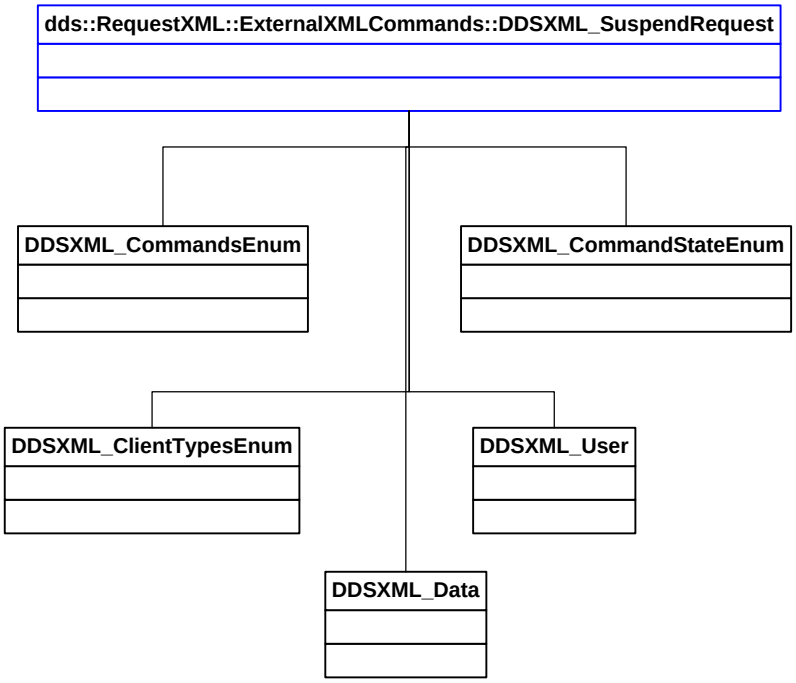

Figure 3.3.11-1, DDSXML_SuspendRequest Class UML Diagram

3.3.11.1 DDSXML_SuspendRequest Functions

3.3.11.1.1 DDSXML_SuspendRequest::DDSXML_SuspendRequest

dds::RequestXML::ExternalXMLCommands::DDSXML_SuspendRequest::DDSXML_SuspendRequest (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.11.1.2 DDSXML_SuspendRequest::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_SuspendRequest::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.12 DDSXML_SuspendRequestResponse Class Reference

This class is responsible for handling the Suspend Request Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a suspend request response command

The Class diagram representing the DDSXML_SuspendRequestResponse Class is provided in Figure 3.3.12-1, DDSXML_SuspendRequestResponse Class UML Diagram.

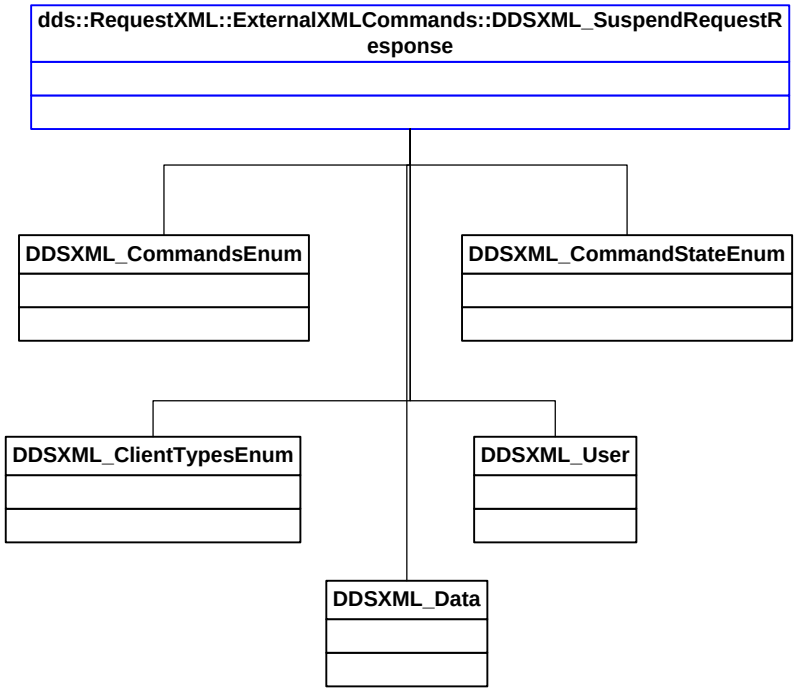

Figure 3.3.12-1, DDSXML_SuspendRequestResponse Class UML Diagram

3.3.12.1 DDSXML_SuspendRequestResponse Functions

3.3.12.1.1 DDSXML_SuspendRequestResponse::DDSXML_SuspendRequestResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_SuspendRequestResponse::DDSXML_SuspendRequestResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.12.1.2 DDSXML_SuspendRequestResponse::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_SuspendRequestResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.13 DDSXML_ResumeRequest Class Reference

This class is responsible for handling the Resume Request command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a resume request command

The Class diagram representing the DDSXML_ResumeRequest Class is provided in Figure 3.3.13-1, DDSXML_ResumeRequest Class UML Diagram.

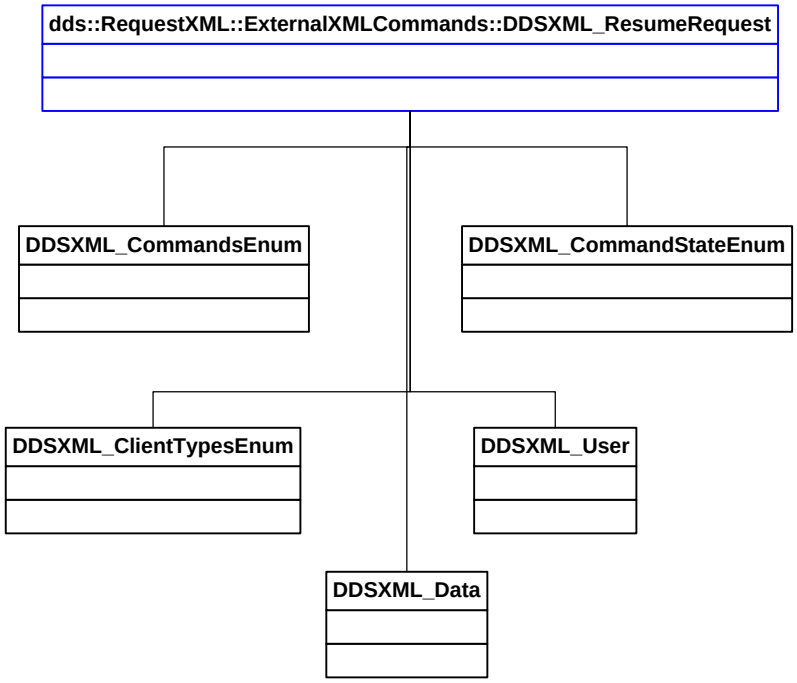

Figure 3.3.13-1, DDSXML_ResumeRequest Class UML Diagram

3.3.13.1 DDSXML_ResumeRequest Functions

3.3.13.1.1 DDSXML_ResumeRequest::DDSXML_ResumeRequest

dds::RequestXML::ExternalXMLCommands::DDSXML_ResumeRequest::DDSXML_ResumeRequest (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.13.1.2 DDSXML_ResumeRequest::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_ResumeRequest::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.14 DDSXML_ResumeRequestResponse Class Reference

This class is responsible for handling the Resume Request Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a resume request response command

The Class diagram representing the DDSXML_ResumeRequestResponse Class is provided in Figure 3.3.14-1, DDSXML_ResumeRequestResponse Class UML Diagram.

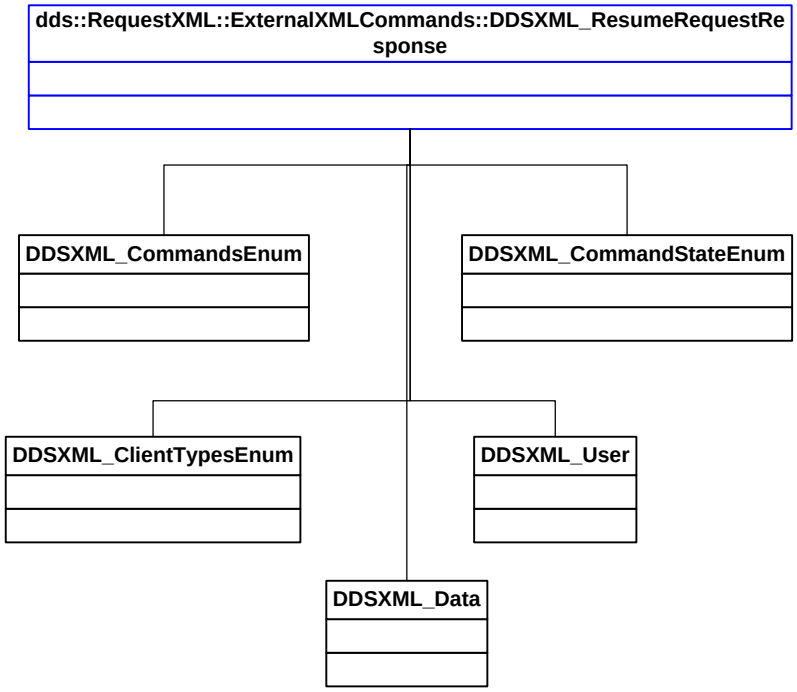

Figure 3.3.14-1, DDSXML_ResumeRequestResponse Class UML Diagram

3.3.14.1 DDSXML_ResumeRequestResponse Functions

3.3.14.1.1 DDSXML_ResumeRequestResponse::DDSXML_ResumeRequestResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_ResumeRequestResponse::DDSXML_ResumeRequestResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.14.1.2 DDSXML_ResumeRequestResponse::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_ResumeRequestResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.15 DDSXML_GetRequestByID Class Reference

This class is responsible for handling the Get Request by ID command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get request by ID command

The Class diagram representing the DDSXML_GetRequestByID Class is provided in Figure 3.3.15-1, DDSXML_GetRequestByID Class UML Diagram.

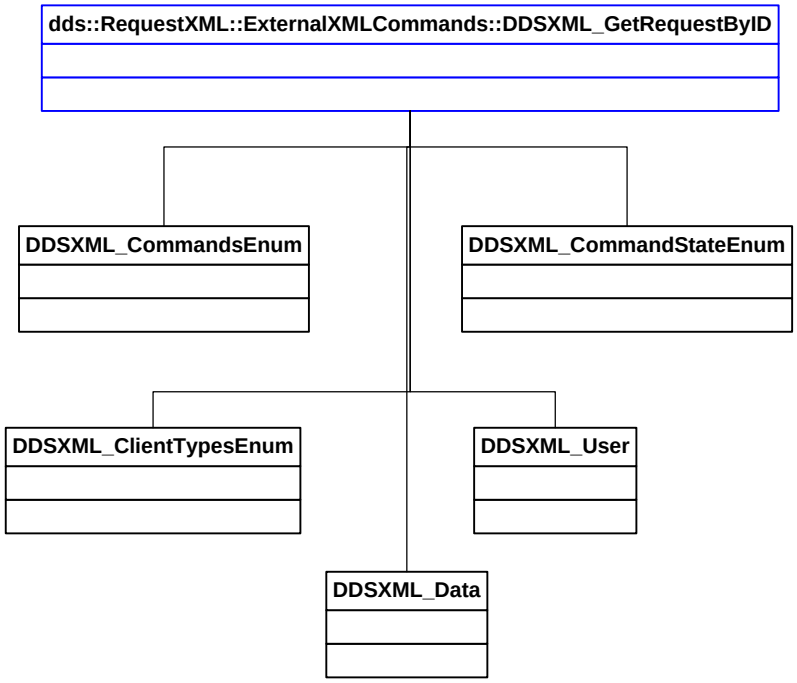

Figure 3.3.15-1, DDSXML_GetRequestByID Class UML Diagram

3.3.15.1 DDSXML_GetRequestByID Functions

3.3.15.1.1 DDSXML_GetRequestByID::DDSXML_GetRequestByID

dds::RequestXML::ExternalXMLCommands::DDSXML_GetRequestByID::DDSXML_GetRequestByID (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.15.1.2 DDSXML_GetRequestByID::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_GetRequestByID::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.16 DDSXML_GetRequestByIDResponse Class Reference

This class is responsible for handling the Get Request by ID Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class enables a user to build a get request by ID response command

The Class diagram representing the DDSXML_GetRequestByIDResponse Class is provided in Figure 3.3.16-1, DDSXML_GetRequestByIDResponse Class UML Diagram.

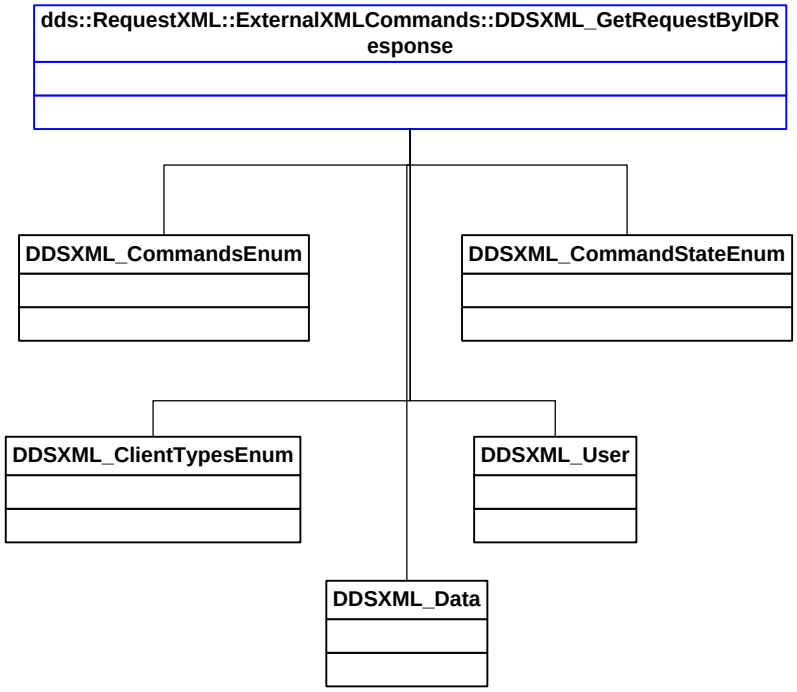

Figure 3.3.16-1, DDSXML_GetRequestByIDResponse Class UML Diagram

3.3.16.1 DDSXML_GetRequestByIDResponse Functions

3.3.16.1.1 DDSXML_GetRequestByIDResponse::DDSXML_GetRequestByIDRespon se

dds::RequestXML::ExternalXMLCommands::DDSXML_GetRequestByIDResponse::D DSXML_GetRequestByIDResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.16.1.2 DDSXML_GetRequestByIDResponse::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_GetRequestByIDResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.17 DDSXML_JMSRetransmit Class Reference

This class is responsible for handling the JMS Retransmit command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a retransmit request command

The Class diagram representing the DDSXML_JMSRetransmit Class is provided in Figure 3.3.17-1, DDSXML_JMSRetransmit Class UML Diagram.

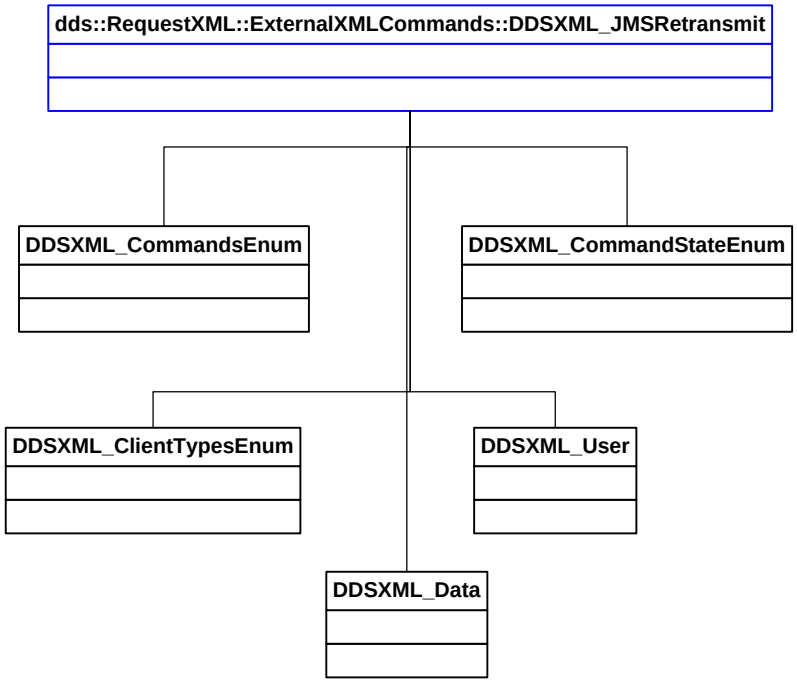

Figure 3.3.17-1, DDSXML_JMSRetransmit Class UML Diagram

3.3.17.1 DDSXML_JMSRetransmit Functions

3.3.17.1.1 DDSXML_JMSRetransmit::DDSXML_JMSRetransmit

dds::RequestXML::ExternalXMLCommands::DDSXML_JMSRetransmit::DDSXML_JMSRetransmit (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.17.1.2 DDSXML_JMSRetransmit::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_JMSRetransmit::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_RetransmitRequest

Returns:

- The DDSXML_RetransmitRequest object or null if none.

3.3.18 DDSXML_JMSRetransmitResponse Class Reference

This class is responsible for handling the JMS Retransmit Request Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class enables a user to build a JMS retransmit request response command

The Class diagram representing the DDSXML_JMSRetransmitResponse Class is provided in Figure 3.3.18-1, DDSXML_JMSRetransmitResponse Class UML Diagram.

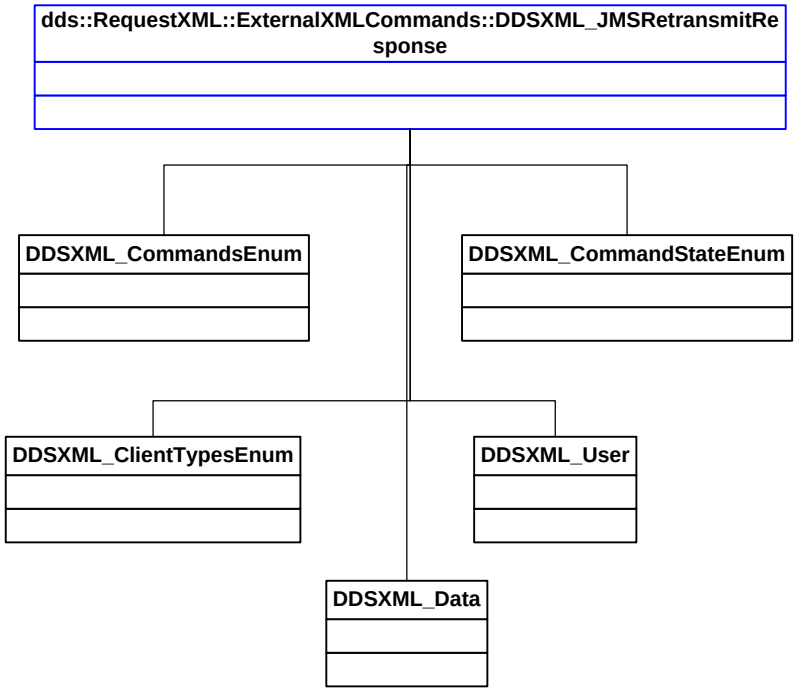

Figure 3.3.18-1, DDSXML_JMSRetransmitResponse Class UML Diagram

3.3.18.1 DDSXML_JMSRetransmitResponse Functions

3.3.18.1.1 DDSXML_JMSRetransmitResponse::DDSXML_JMSRetransmitResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_JMSRetransmitResponse::DDSXML_JMSRetransmitResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.18.1.2 DDSXML_JMSRetransmitResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_JMSRetransmitResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Request

Returns:

- The DDSXML_Request object or null if none.

3.3.19 DDSXML_CreateDestination Class Reference

This class is responsible for handling the Create Destination command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a create destination command

The Class diagram representing the DDSXML_CreateDestination Class is provided in Figure 3.3.19-1, DDSXML_CreateDestination Class UML Diagram.

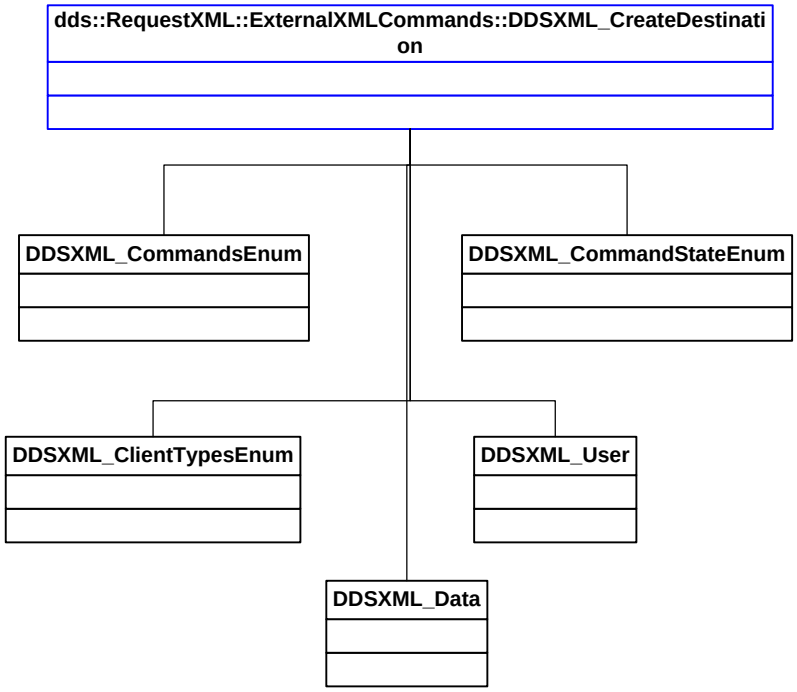

Figure 3.3.19-1, DDSXML_CreateDestination Class UML Diagram

3.3.19.1 DDSXML_CreateDestination Functions

3.3.19.1.1 DDSXML_CreateDestination::DDSXML_CreateDestination

dds::RequestXML::ExternalXMLCommands::DDSXML_CreateDestination::DDSXML_CreateDestination (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- Command The command to execute on the Server
- clientType The Type of client for this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.

3.3.19.1.2 DDSXML_CreateDestination::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_CreateDestination::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_Destination

Returns:

- The DDSXML_Destination object or null if none.

3.3.20 DDSXML_CreateDestinationResponse Class Reference

This class is responsible for handling the Create Destination Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Request Server to the Java API. This class contains the Request Server response to a user create destination command

The Class diagram representing the DDSXML_CreateDestinationResponse Class is provided in Figure 3.3.20-1, DDSXML_CreateDestinationResponse Class UML Diagram.

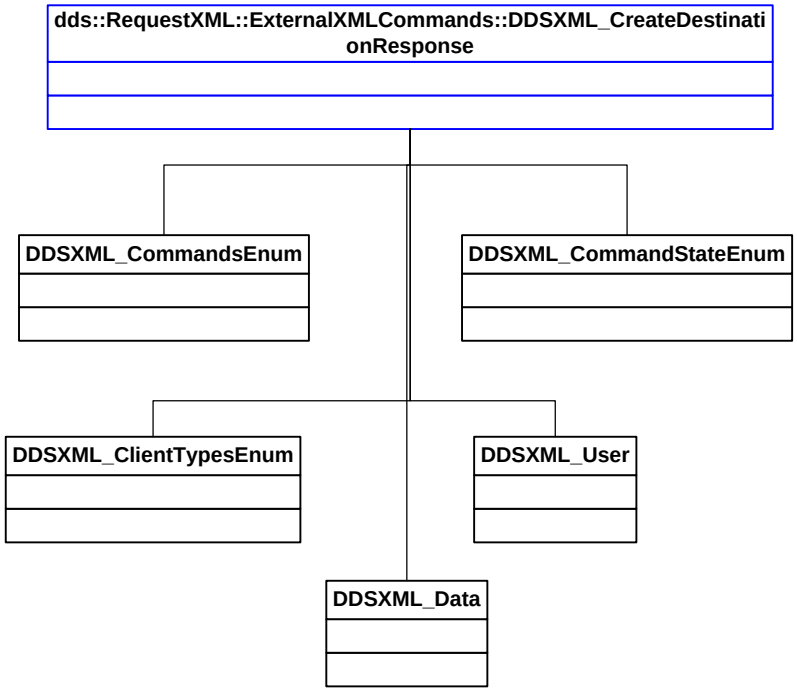

Figure 3.3.20-1, DDSXML_CreateDestinationResponse Class UML Diagram

3.3.20.1 DDSXML_CreateDestinationResponse Functions

3.3.20.1.1 DDSXML_CreateDestinationResponse::DDSXML_CreateDestinationResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_CreateDestinationResponse::DDSXML_CreateDestinationResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.20.1.2 DDSXML_CreateDestinationResponse::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_CreateDestinationResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.21 DDSXML_GetDestinations Class Reference

This class is responsible for handling the Get Destinations command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get destinations command

The Class diagram representing the DDSXML_GetDestinations Class is provided in Figure 3.3.21-1, DDSXML_GetDestinations Class UML Diagram.

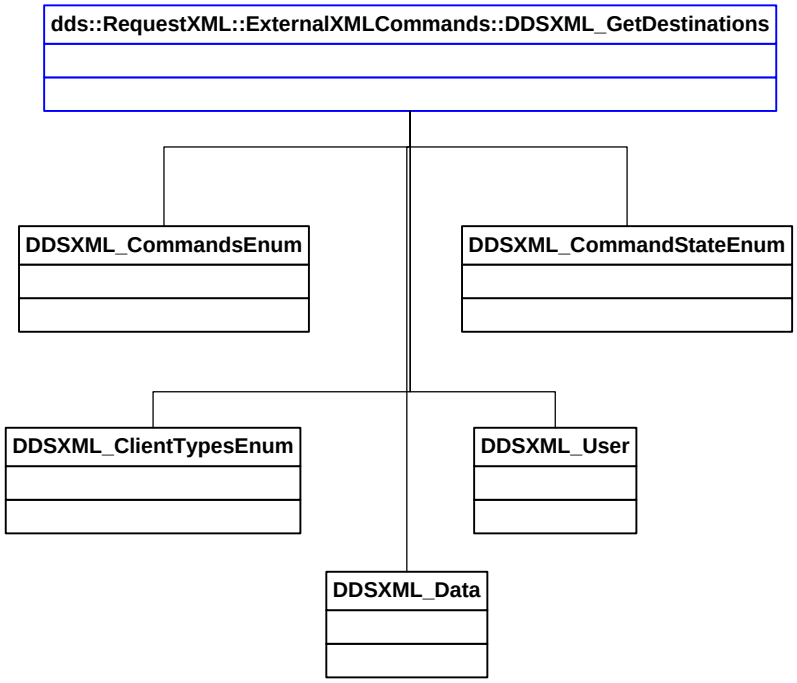

Figure 3.3.21-1, DDSXML_GetDestinations Class UML Diagram

3.3.21.1 DDSXML_GetDestinations Functions

3.3.21.1.1 DDSXML_GetDestinations::DDSXML_GetDestinations

dds::RequestXML::ExternalXMLCommands::DDSXML_GetDestinations::DDSXML_GetDestinations (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.21.1.2 DDSXML_GetDestinations::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_GetDestinations::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_DestinationList

Returns:

- The DDSXML_DestinationList object or null if none.

3.3.22 DDSXML_GetDestinations Class Reference

This class is responsible for handling the Get Destinations Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get destinations response command

The Class diagram representing the DDSXML_GetDestinationsResponse Class is provided in Figure 3.3.22-1, DDSXML_GetDestinationsResponse Class UML Diagram.

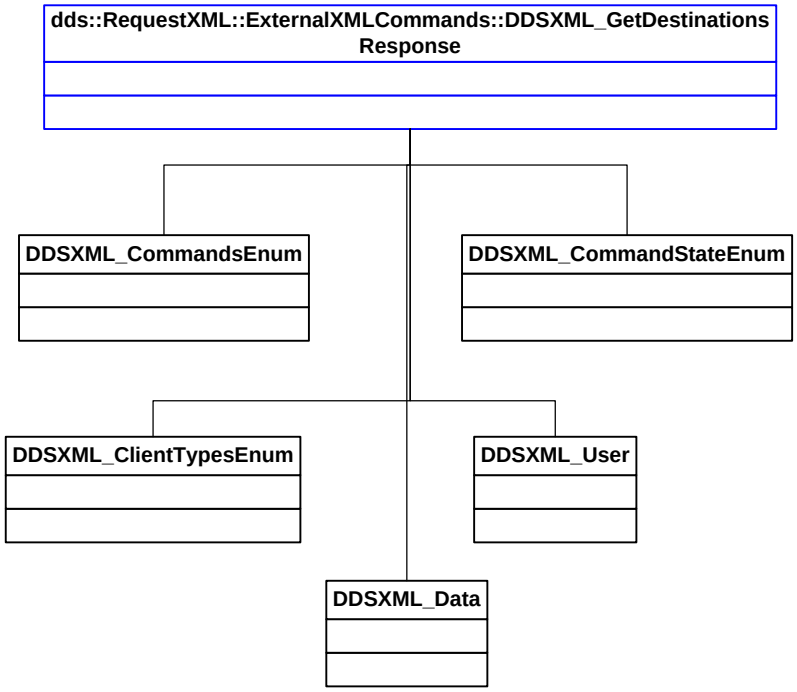

Figure 3.3.22-1, DDSXML_GetDestinationsResponse Class UML Diagram

3.3.22.1 DDSXML_GetDestinationsResponse Functions

3.3.22.1.1 DDSXML_GetDestinationsResponse::DDSXML_GetDestinationsRespons e

dds::RequestXML::ExternalXMLCommands::DDSXML_GetDestinationsResponse::DDSXML_GetDestinationsResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.22.1.2 DDSXML_GetDestinationsResponse::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_GetDestinationsResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_DestinationList

Returns:

- The DDSXML_DestinationList object or null if none.

3.3.23 DDSXML_GetDataProducts Class Reference

This class is responsible for handling the Get Data Productss command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get data products command

The Class diagram representing the DDSXML_GetDataProducts Class is provided in Figure 3.3.23-1, DDSXML_GetDataProducts Class UML Diagram.

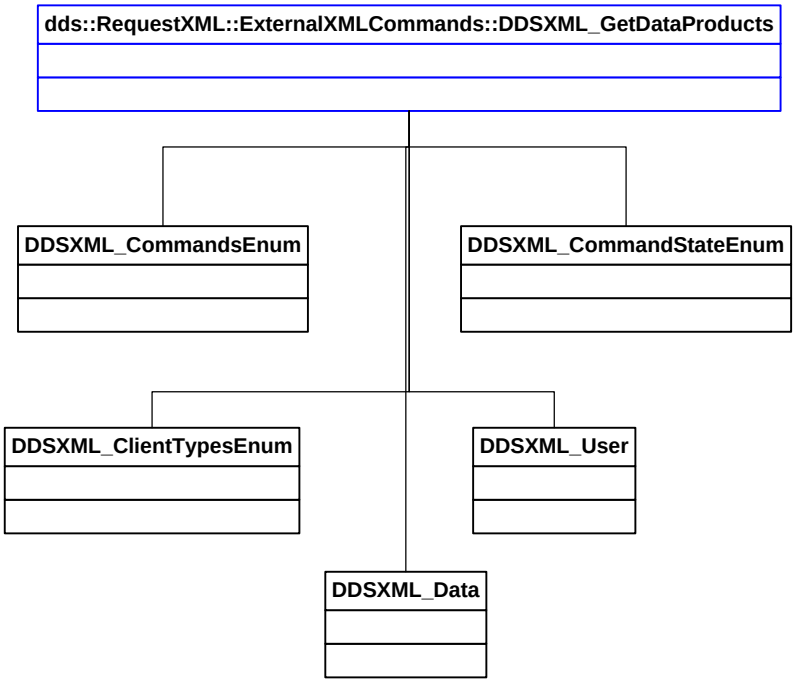

Figure 3.3.23-1, DDSXML_GetDataProducts Class UML Diagram

3.3.23.1 DDSXML_GetDataProducts Functions

3.3.23.1.1 DDSXML_GetDataProducts::DDSXML_GetDataProducts

dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataProducts::DDSXML_GetDataProducts (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.23.1.2 DDSXML_GetDataProducts::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataProducts::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_NoData

Returns:

- The DDSXML_NoData object or null if none.

3.3.24 DDSXML_GetDataProductsResponse Class Reference

This class is responsible for handling the Get Data Products Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get data products response command

The Class diagram representing the DDSXML_GetDataProductsResponse Class is provided in Figure 3.3.24-1, DDSXML_GetDataProductsResponse Class UML Diagram.

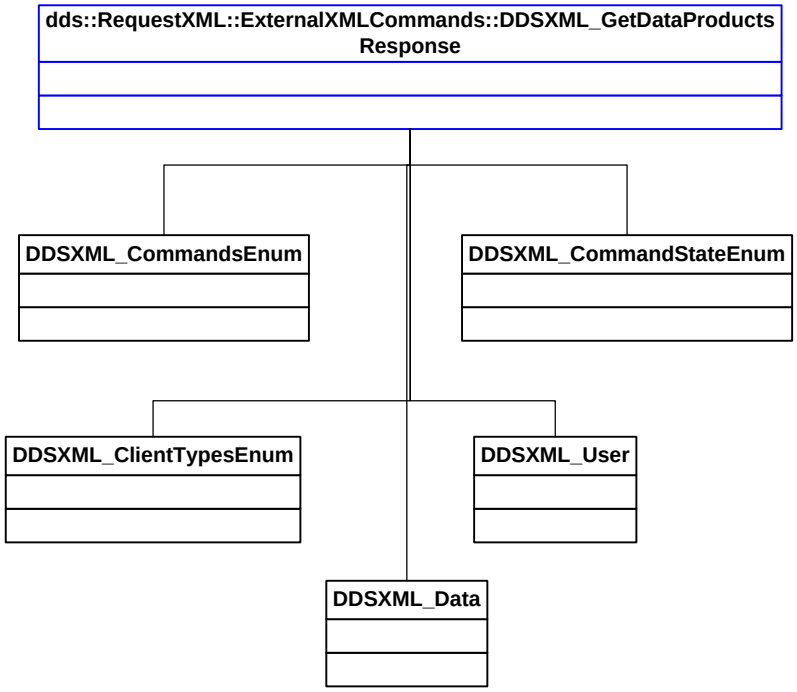

Figure 3.3.24-1, DDSXML_GetDataProductsResponse Class UML Diagram

3.3.24.1 DDSXML_GetDataProductsResponse Functions

3.3.24.1.1 DDSXML_GetDataProductsResponse::DDSXML_GetDataProductsRespo nse

dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataProductsResponse::D DSXML_GetDataProductsResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.24.1.2 DDSXML_GetDataProductsResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataProductsResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_DataProductList

Returns:

- The DDSXML_DataProductList object or null if none.

3.3.25 DDSXML_GetDataShipmentList Class Reference

This class is responsible for handling the Get Data Shipment List command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get data shipment list command

The Class diagram representing the DDSXML_GetDataShipmentList Class is provided in Figure 3.3.25-1, DDSXML_GetDataShipmentList Class UML Diagram.

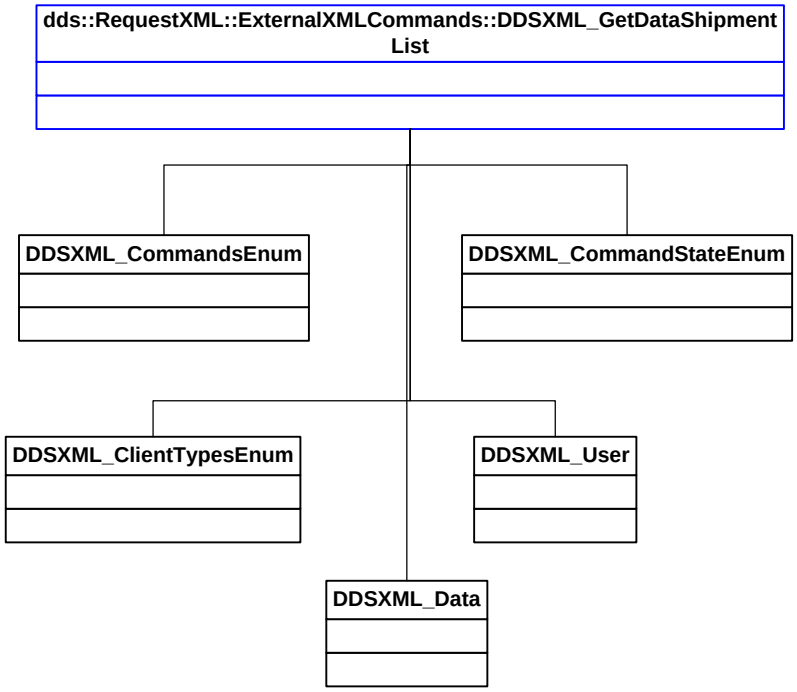

Figure 3.3.25-1, DDSXML_GetDataShipmentList Class UML Diagram

3.3.25.1 DDSXML_GetDataShipmentList Functions

3.3.25.1.1 DDSXML_GetDataShipmentList::DDSXML_GetDataShipmentList

dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataShipmentList::DDSXML_GetDataShipmentList (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.25.1.2 DDSXML_GetDataShipmentList::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataShipmentList::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_DataID

Returns:

- The DDSXML_DataID object or null if none.

3.3.26 DDSXML_JMSSubscribeToSystemMessages Class Reference

This class is responsible for handling the command XML to request the JMS API push status messages to a client JMS Topic. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from a JMS Client to the JMS API. This class enables a user to build a JMS subscribe to system messages command

The Class diagram representing the DDSXML_JMSSubscribeToSystemMessages Class is provided in Figure 3.3.26-1, DDSXML_JMSSubscribeToSystemMessages Class UML Diagram.

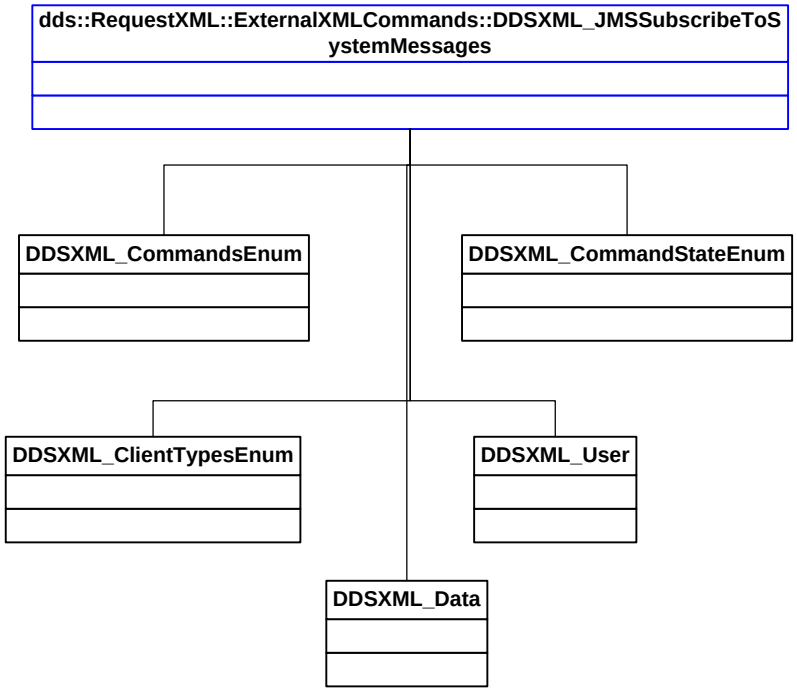

Figure 3.3.26-1, DDSXML_JMSSubscribeToSystemMessages Class UML Diagram

3.3.26.1 DDSXML_JMSSubscribeToSystemMessages Functions

3.3.26.1.1 DDSXML_JMSSubscribeToSystemMessages::DDSXML_JMSSubscribeToSystemMessages

dds::RequestXML::ExternalXMLCommands::DDSXML_JMSSubscribeToSystemMessages::DDSXML_JMSSubscribeToSystemMessages (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.26.1.2 DDSXML_JMSSubscribeToSystemMessages::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_JMSSubscribeToSystemMessages::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_JmsSubscriber

Returns:

- The DDSXML_JmsSubscriber object or null if none.

3.3.27 DDSXML_JMSSubscribeToSystemMessagesResponse Class Reference

This class is responsible for handling the response command XML for the request to the JMS API to push status messages to a client JMS Topic. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from a JMS API to a JMS Client. This class enables a userthe JMS API to build a JMS subscribe to system messages response command

The Class diagram representing the DDSXML_JMSSubscribeToSystemMessagesResponse Class is provided in Figure 3.3.27-1, DDSXML_JMSSubscribeToSystemMessagesResponse Class UML Diagram.

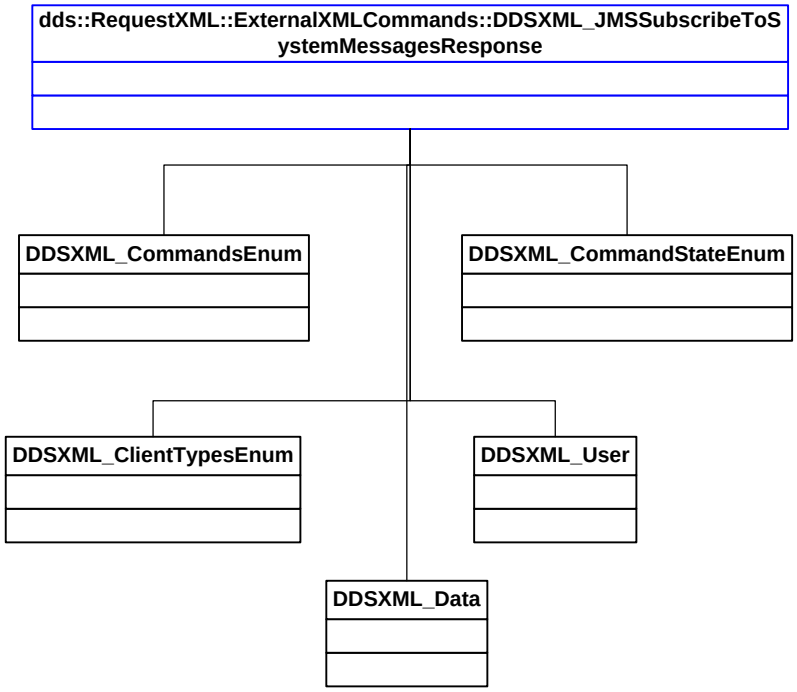

Figure 3.3.27-1, DDSXML_JMSSubscribeToSystemMessagesResponse Class UML Diagram

3.3.27.1 DDSXML_JMSSubscribeToSystemMessagesResponse Functions

3.3.27.1.1 DDSXML_JMSSubscribeToSystemMessagesResponse::DDSXML_JMSSubscribeToSystemMessagesResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_JMSSubscribeToSystemMessagesResponse::DDSXML_JMSSubscribeToSystemMessagesResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.
- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.27.1.2 DDSXML_JMSSubscribeToSystemMessagesResponse::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_JMSSubscribeToSystemMessagesResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_NoData

Returns:

- The DDSXML_NoData object or null if none.

3.3.28 DDSXML_JMSStopSubscribing Class Reference

This class is responsible for handling the command XML to request the JMS API stop pushing status messages to a client JMS Topic. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from a JMS Client to the JMS API. This class enables a user to build a JMS stop subscribing to system messages command

The Class diagram representing the DDSXML_JMSStopSubscribing Class is provided in Figure 3.3.28-1, DDSXML_JMSStopSubscribing Class UML Diagram.

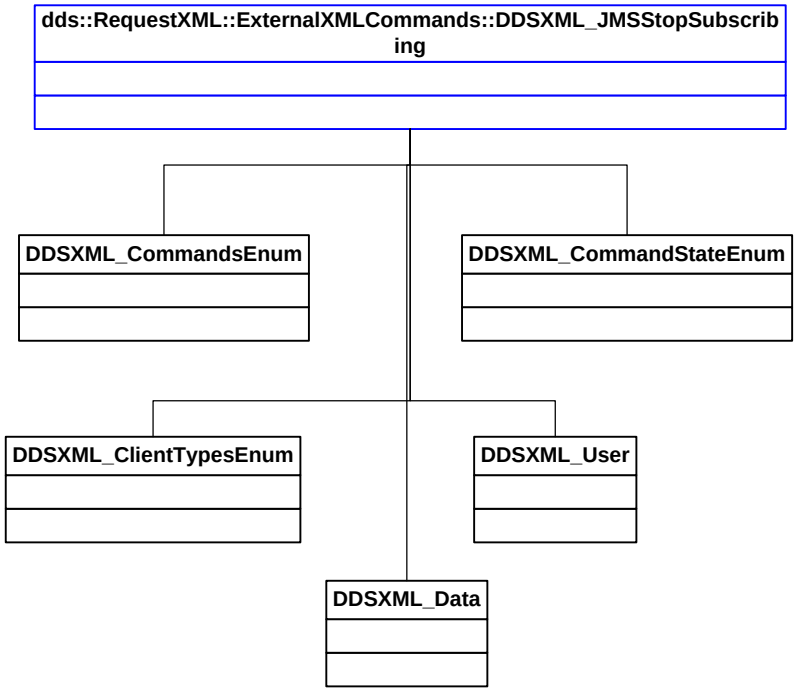

Figure 3.3.28-1, DDSXML_JMSStopSubscribing Class UML Diagram

3.3.28.1 DDSXML_JMSStopSubscribing Functions

3.3.28.1.1 DDSXML_JMSStopSubscribing::DDSXML_JMSStopSubscribing

dds::RequestXML::ExternalXMLCommands::DDSXML_JMSStopSubscribing::DDSXML_JMSStopSubscribing (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.28.1.2 DDSXML_JMSStopSubscribing::getCommandDataFromXML

DDSXML_Data
dds::RequestXML::ExternalXMLCommands::DDSXML_JMSStopSubscribing::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.29 DDSXML_JMSStopSubscribingResponse Class Reference

This class is responsible for handling the response command XML for the request to the JMS API to stop pushing status messages to a client JMS Topic. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the JMS API to a JMS Client. This class enables the JMS API to build a JMS stop subscribing to system messages response command

The Class diagram representing the DDSXML_JMSStopSubscribingResponse Class is provided in Figure 3.3.29-1, DDSXML_JMSStopSubscribingResponse Class UML Diagram.

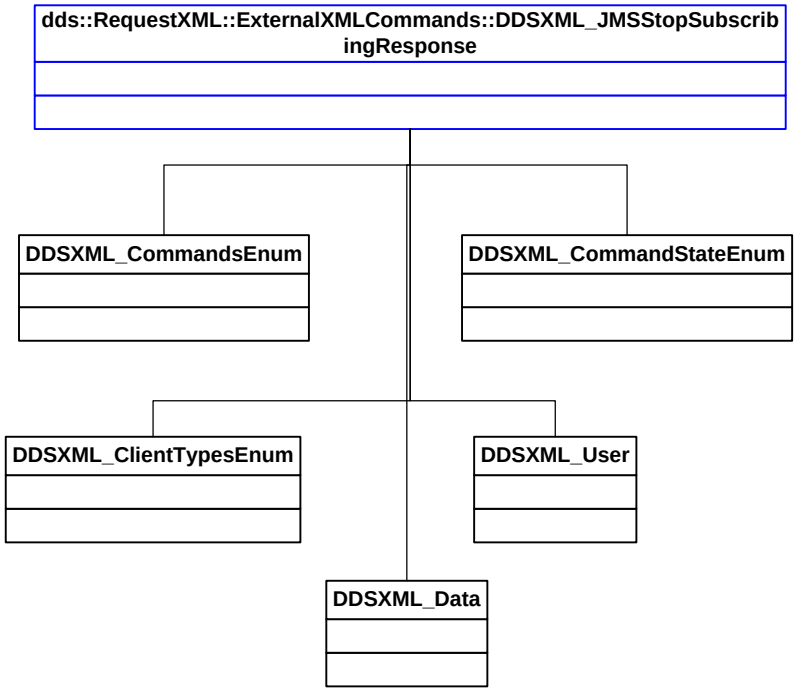

Figure 3.3.29-1, DDSXML_JMSStopSubscribingResponse Class UML Diagram

3.3.29.1 DDSXML_JMSStopSubscribingResponse Functions

3.3.29.1.1 DDSXML_JMSStopSubscribingResponse::DDSXML_JMSStopSubscribingResponse

dds::RequestXML::ExternalXMLCommands::DDSXML_JMSStopSubscribingResponse::DDSXML_JMSStopSubscribingResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.29.1.2 DDSXML_JMSStopSubscribingResponse::getCommandDataFromXML

DDSXML_Data dds::RequestXML::ExternalXMLCommands::DDSXML_JMSStopSubscribingResponse::getCommandDataFromXML (

)

This method retrieves the command data from the XML. This command does not have command data.

Returns:

- There is no command data for this command, it always returns null.

3.3.30 DDSXML_GetRequestIDs Class Reference

This class is responsible for handling the Get Request IDs command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get request IDs command.

The Class diagram representing the DDSXML_GetRequestIDs Class is provided in Figure 3.3.30-1, DDSXML_GetRequestIDs Class UML Diagram.

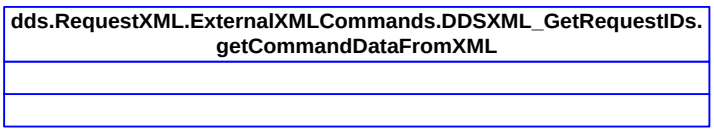

Figure 3.3.30-1, DDSXML_GetRequestIDs Class UML Diagram

3.3.30.1 DDSXML_GetRequestIDs Functions

3.3.30.1.1 DDSXML_GetRequestIDs.getCommandDataFromXML

DDSXML_Data dds.RequestXML.ExternalXMLCommands.DDSXML_GetRequestIDs.getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_NoData

Returns:

- The DDSXML_NoData object or null if none.

3.3.31 DDSXML_GetRequestIDsResponse Class Reference

This class is responsible for handling the Get Request IDs Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get request IDs response command

The Class diagram representing the DDSXML_GetRequestIDsResponse Class is provided in Figure 3.3.31-1, DDSXML_GetRequestIDsResponse Class UML Diagram.

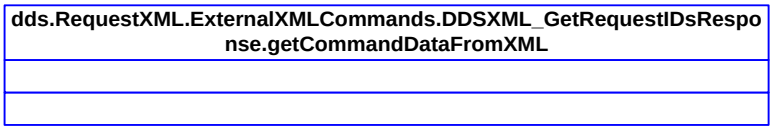

Figure 3.3.31-1, DDSXML_GetRequestIDsResponse Class UML Diagram

3.3.31.1 DDSXML_GetRequestIDsResponse Functions

3.3.31.1.1 DDSXML_GetRequestIDsResponse.getCommandDataFromXML

DDSXML_Data
dds.RequestXML.ExternalXMLCommands.DDSXML_GetRequestIDsResponse.getCommandDataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_RequestIDList

Returns:

- The DDSXML_RequestIDList object or null if none.

3.3.32 DDSXML_DeleteDestinations Class Reference

This class is responsible for handling the Delete Destination XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. These classes are contain the commands that are sent between the Request Server and the API. All command XML reside in these classes and not in the API or Request Server.

The Class diagram representing the DDSXML_DeleteDestinations Class is provided in Figure 3.3.32-1, DDSXML_DeleteDestinations Class UML Diagram.

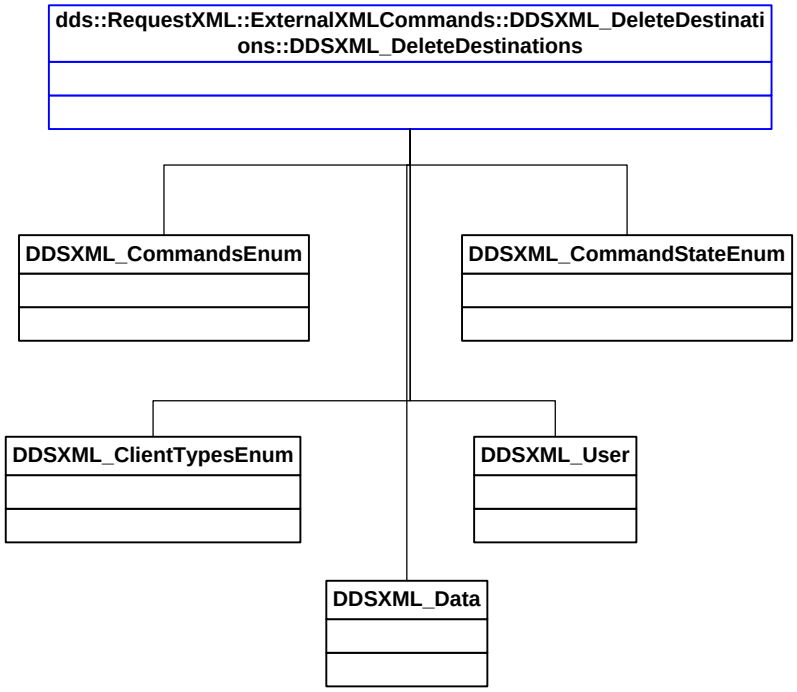

Figure 3.3.32-1, DDSXML_DeleteDestinations Class UML Diagram

3.3.32.1 DDSXML_DeleteDestinations Functions

dds::RequestXML::ExternalXMLCommands::DDSXML_DeleteDestinations::DDSXML_DeleteDestinations (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.33 DDSXML_DeleteDestinationsResponse Class Reference

This class is responsible for handling the Delete Destination Response XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. These classes are contain the commands that are sent between the Request Server and the API. All command XML reside in these classes and not in the API or Request Server.

The Class diagram representing the DDSXML_DeleteDestinationsResponse Class is provided in Figure 3.3.33-1, DDSXML_DeleteDestinationsResponse Class UML Diagram.

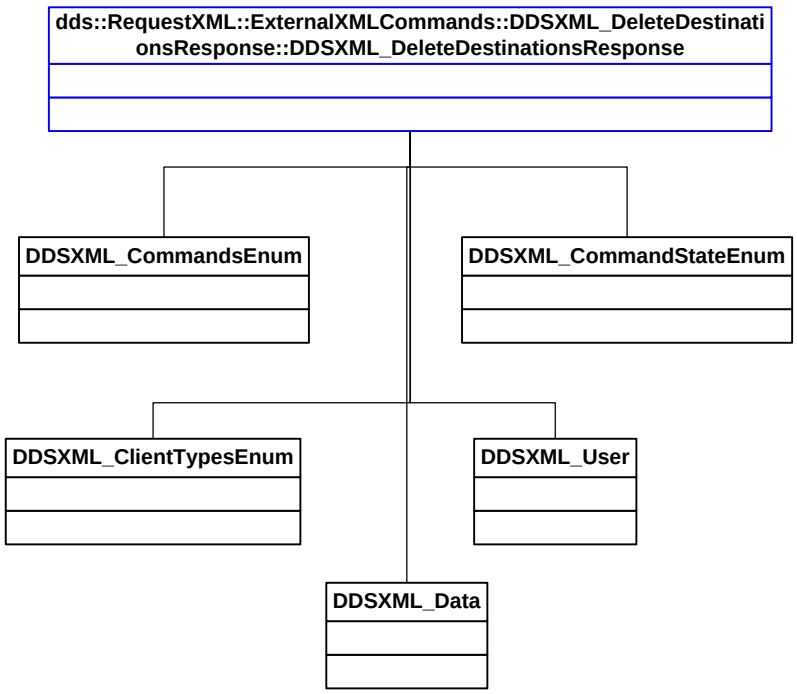

Figure 3.3.33-1, DDSXML_DeleteDestinationsResponse Class UML Diagram

3.3.33.1 DDSXML_DeleteDestinationsResponse Functions

dds::RequestXML::ExternalXMLCommands::DDSXML_DeleteDestinationsResponse::
DDSXML_DeleteDestinationsResponse (

```
    String connectionID,
    String commandID,
    DDSXML_CommandsEnum command,
    DDSXML_CommandStateEnum commandState,
    String commandData,
    DDSXML_ClientTypesEnum clientType,
    DDSXML_User user,
    String dataID
  )
```

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.
- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.34 DDSXML_GetStatusMessages Class Reference

This class is responsible for handling the Get Staus Messages Response command XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. This class encapsulates the XML for a command sent from the Java API to the Request Server. This class enables a user to build a get status messages response command

The Class diagram representing the DDSXML_GetStatusMessages Class is provided in Figure 3.3.34-1, DDSXML_GetStatusMessages Class UML Diagram.

dds.RequestXML.ExternalXMLCommands.DDSXML_GetStatusMessages. getCommandDataFromXML

Figure 3.3.34-1, DDSXML_GetStatusMessages Class UML Diagram

3.3.34.1 DDSXML_GetStatusMessages Functions

3.3.34.1.1 DDSXML_GetStatusMessages.getCommandDataFromXML

DDSXML_Data
dds.RequestXML.ExternalXMLCommands.DDSXML_GetStatusMessages.getCommand
DataFromXML (

)

This method retrieves the command data from the XML. The data is returned as the base data class DDSXML_Data but the returned object can be cast to the proper type DDSXML_NoData

Returns:

- The DDSXML_NoData object or null if none.

3.3.35 DDSXML_ GetStatusMessageResponse Class Reference

This class is responsible for handling the Get Status Message Response XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. These classes are contain the commands that are sent between the Request Server and the API. All command XML reside in these classes and not in the API or Request Server.

The Class diagram representing the DDSXML_ GetStatusMessageResponse Class is provided in Figure 3.3.35-1, DDSXML_ GetStatusMessageResponse Class UML Diagram.

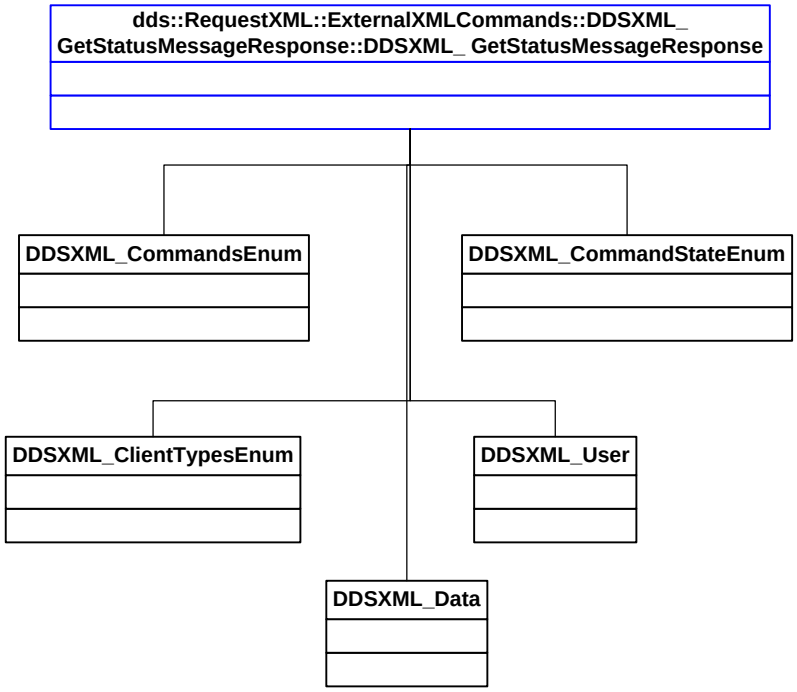

Figure 3.3.35-1, DDSXML_ GetStatusMessageResponse Class UML Diagram

3.3.35.1 DDSXML_ GetStatusMessageResponse Functions

dds::RequestXML::ExternalXMLCommands::DDSXML_
GetStatusMessageResponse::DDSXML_ GetStatusMessageResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.

- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.36 DDSXML_GetDataShipmentListResponse Class Reference

This class is responsible for handling the Get Data Shipment List Response XML. It creates, reads, writes and extracts the data from/to the XML. It also allows for validation of the data. If the data and XML is validated then no invalid XML should be sent across the API. These classes are contain the commands that are sent between the Request Server and the API. All command XML reside in these classes and not in the API or Request Server.

The Class diagram representing the DDSXML_GetDataShipmentListResponse Class is provided in Figure 3.3.36-1, DDSXML_GetDataShipmentListResponse Class UML Diagram.

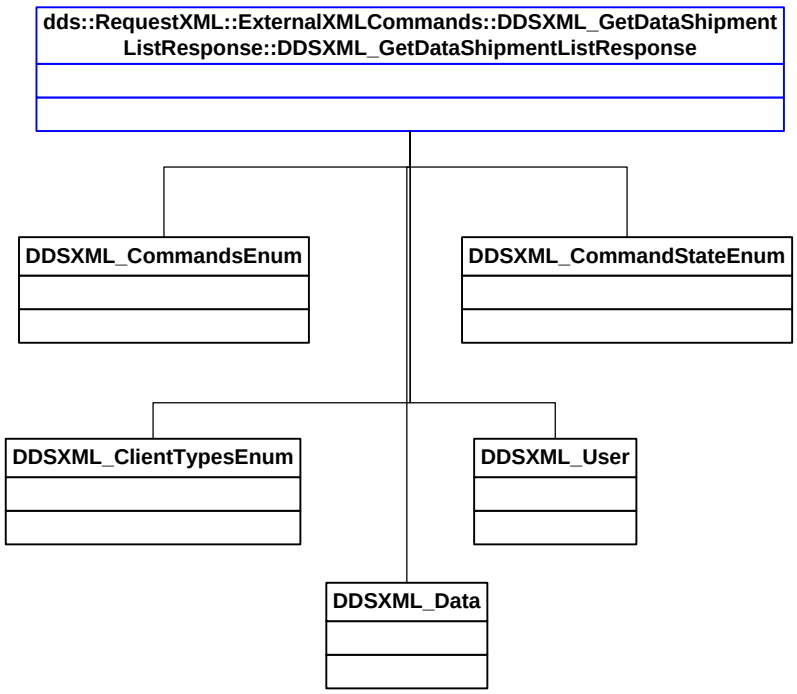

Figure 3.3.36-1, DDSXML_GetDataShipmentListResponse Class UML Diagram

3.3.36.1 DDSXML_GetDataShipmentListResponse Functions

dds::RequestXML::ExternalXMLCommands::DDSXML_GetDataShipmentListResponse ::DDSXML_GetDataShipmentListResponse (

String connectionID,

String commandID,

DDSXML_CommandsEnum command,

DDSXML_CommandStateEnum commandState,

String commandData,

DDSXML_ClientTypesEnum clientType,

DDSXML_User user,

String dataID

)

Overloaded constructor that fills in all the command data needed. Note that commandData must be passed to the proper XML object based on the command sent to create an object to use to get the data for that command.

Parameters:

- connectionID The connection ID that sent this command to the Server.
- commandID A unique ID for each command sent to the Server. This ID will be used in the response command
- command The command to execute on the Server
- commandState This is the state of the command
- commandData This is the XML text that is to be used to create the proper Command based on the command sent.
- clientType The Type of client for this command
- user The user object for the owner of this command
- dataID The ID of the data to execute the command on. This can be a Request,Catalog, Query, etc.

3.3.37 DDSXML_CommandsEnum Class Reference

This class is defines all commands from the client to the Request Server and back.

The Class diagram representing the DDSXML_CommandsEnum Class is provided in Figure 3.3.37-1, DDSXML_CommandsEnum Class UML Diagram.

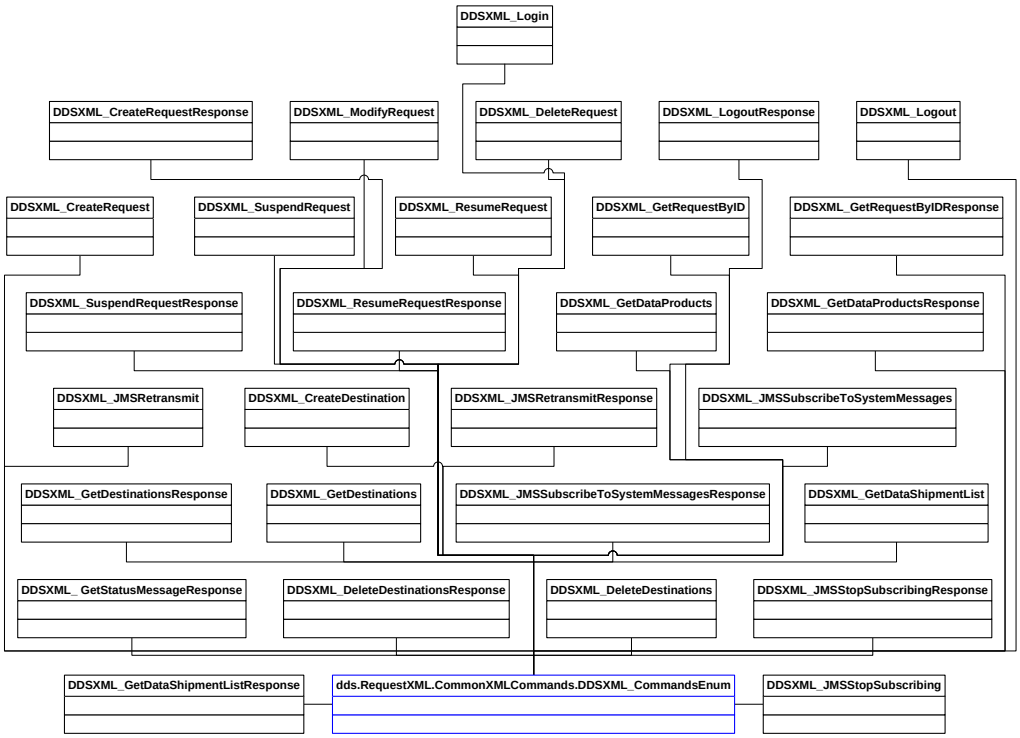

Figure 3.3.37-1, DDSXML_CommandsEnum Class UML Diagram

3.3.37.1 DDSXML_CommandsEnum Attributes

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UNKNOWN_CMD = 0
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UNKNOWN_RESPONSE_CMD = 1
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CONNECT_CMD = 2
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CONNECT_RESPONSE_CMD = 3
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DISCONNECT_CMD = 4
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DISCONNECT_RESPONSE_CMD = 5

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGIN_CMD = 6
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGIN_RESPONSE_CMD = 7
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGOUT_CMD = 8
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGOUT_RESPONSE_CMD = 9
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_STATUS_MESSAGE_CMD = 10
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.STATUS_MESSAGE_RESPONSE_CMD = 11
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_API_HEARTBEAT_CMD = 12
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_ROLE_LIST_CMD = 13
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.ROLE_LIST_RESPONSE_CMD = 14
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_API_CONFIGURATION_CMD = 15
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.API_CONFIGURATION_RESPONSE_CMD = 16
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_GUI_CONFIGURATION_CMD = 17
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GUI_CONFIGURATION_RESPONSE_CMD = 18

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_NEXT_INDEX_CMD = 19
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.NEXT_INDEX_RESPONSE_CMD = 20
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_SUPERVISOR_CMD = 21
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SUPERVISOR_RESPONSE_CMD = 22
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_DESTINATION_CMD = 23
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_DESTINATION_RESPONSE_CMD = 24
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_DESTINATION_CMD = 25
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_DESTINATION_RESPONSE_CMD = 26
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_CMD = 27
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_RESPONSE_CMD = 28
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_BY_ID_CMD = 29
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_BY_ID_RESPONSE_CMD = 30
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DESTINATION_CMD = 31

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DESTINATION_RESPONSE_CMD = 32
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.VALIDATE_DESTINATION_CMD = 33
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.VALIDATE_DESTINATION_RESPONSE_CMD = 34
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_CMD = 35
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_RESPONSE_CMD = 36
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_BY_ID_CMD = 37
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_BY_ID_RESPONSE_CMD = 38
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_REQUEST_CMD = 39
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_REQUEST_RESPONSE_CMD = 40
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_REQUEST_CMD = 41
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_REQUEST_RESPONSE_CMD = 42
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_CMD = 43
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_RESPONSE_CMD = 44

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_REQUEST_CMD = 45
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_REQUEST_RESPONSE_CMD = 46
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RESUME_REQUEST_CMD = 47
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RESUME_REQUEST_RESPONSE_CMD = 48
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SUSPEND_REQUEST_CMD = 49
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SUSPEND_REQUEST_RESPONSE_CMD = 50
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_REQUEST_CMD = 51
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_REQUEST_RESPONSE_CMD = 52
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_REQUEST_CMD = 53
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_TEMPLATE_CMD = 54
- static final int static dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_TEMPLATE_RESPONSE_CMD = 55
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_TEMPLATE_CMD = 56
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_TEMPLATE_RESPONSE_CMD = 57

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_TEMPLATE_CMD = 58
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_TEMPLATE_RESPONSE_CMD = 59
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.IDLE_HANDLER_CMD = 60
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.IDLE_HANDLER_RESPONSE_CMD = 61
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RUN_REQUEST_CMD = 62
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RUN_REQUEST_RESPONSE_CMD = 63
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DATA_SHIPMENT_CMD = 64
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DATA_SHIPMENT_RESPONSE_CMD = 65
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_PROGRESS_CMD = 66
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_PROGRESS_RESPONSE_CMD = 67
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_STATUS_MESSAGE_CMD = 68
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_STATUS_MESSAGE_RESPONSE_CMD = 69
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.STOP_FILE_TRANSFER__CMD = 70

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.STOP_FILE_TRANSFER_RESPONSE_CMD = 71
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TERMINATE_HANDLER_CMD = 72
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TERMINATE_HANDLER_RESPONSE_CMD = 73
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_FILE_CMD = 74
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_FILE_RESPONSE_CMD = 75
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_SHIPMENT_LIST_CMD = 76
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DATA_SHIPMENT_LIST_RESPONSE_CMD = 77
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_CDDR_REQUEST_CMD = 78
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_CDDR_REQUEST_RESPONSE_CMD = 79
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_CDDR_REQUEST_CMD = 80
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_CDDR_REQUEST_RESPONSE_CMD = 81
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_CDDR_REQUEST_CMD = 82
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_CDDR_REQUEST_RESPONSE_CMD = 83

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_CDDR_REQUEST_CMD = 84
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_CDDR_REQUEST_RESPONSE_CMD = 85
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DENY_DATA_LIST_CMD = 86
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DENY_DATA_LIST_RESPONSE_CMD = 87
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_HANDLER_ERROR_CMD = 88
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_HANDLER_STATUS_CMD = 89
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_HANDLER_STATUS_RESPONSE_CMD = 90
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_REQUESTS_CMD = 91
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_REQUESTS_RESPONSE_CMD = 92
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_DESTINATION_CMD = 93
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_SUPERVISOR_CMD = 94
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_BY_ID_CMD = 95
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_BY_ID_RESPONSE_CMD = 96

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_RETRANSMIT_REQUEST_CMD = 97
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_RETRANSMIT_REQUEST_RESPONSE_CMD = 98
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_IDS_CMD = 99
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_IDS_RESPONSE_CMD = 100
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_CDDR_REQUESTS_CMD = 101
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_CDDR_REQUESTS_RESPONSE_CMD = 102
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_CDDR_REQUEST_CMD = 103
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_NEW_REQUEST_CMD = 104
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DENY_DATA_CMD = 105
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DENY_DATA_RESPONSE_CMD = 106
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_DENY_DATA_CMD = 107
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_MASTER_DESTINATIONS_CMD = 108
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_MASTER_DESTINATIONS_RESPONSE_CMD = 109

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_FILTERED_DP_LIST_CMD = 110
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_SYSTEM_MESSAGES_CMD = 111
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DENY_DATA_CMD = 112
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DENY_DATA_RESPONSE_CMD = 113
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_SUBSCRIBE_TO_STATUS_CMD = 114
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_SUBSCRIBE_TO_STATUS_RESPONSE_CMD = 115
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_STOP_SUBSCRIBE_TO_STATUS_CMD = 116
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_STOP_SUBSCRIBE_TO_STATUS_RESPONSE_CMD = 117
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DDR_CMD_ENUM = 118
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MIN_VALUE = UNKNOWN_CMD
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MAX_VALUE = SEND_DDR_CMD + 1

3.3.37.2 DDSXML_CommandsEnum Enumerations

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UNKNOWN_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("UNKNOWN_CMD", UNKNOWN_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UNKNOWN_RESPONSE_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("UNKNOWN_RESPONSE_CMD", UNKNOWN_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CONNECT_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("CONNECT_CMD", CONNECT_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CONNECT_RESPONSE_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("CONNECT_RESPONSE_CMD", CONNECT_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DISCONNECT_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("DISCONNECT_CMD", DISCONNECT_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DISCONNECT_RESPONSE_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("DISCONNECT_RESPONSE_CMD", DISCONNECT_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGIN_CMD_ENUM – Initial value: new DDSXML_CommandsEnum("LOGIN_CMD", LOGIN_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGIN_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("LOGIN_RESPONSE_CMD", LOGIN_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGOUT_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("LOGOUT_CMD", LOGOUT_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.LOGOUT_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("LOGOUT_RESPONSE_CMD", LOGOUT_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_STATUS_MESSAGE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_STATUS_MESSAGE_CMD", GET_STATUS_MESSAGE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.STATUS_MESSAGE_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("STATUS_MESSAGE_RESPONSE_CMD", STATUS_MESSAGE_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_API_HEARTBEAT_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_API_HEARTBEAT_CMD", SEND_API_HEARTBEAT_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_ROLE_LIST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_ROLE_LIST_CMD", GET_ROLE_LIST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.ROLE_LIST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("ROLE_LIST_RESPONSE_CMD", ROLE_LIST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_API_CONFIGURATION_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_API_CONFIGURATION_CMD", GET_API_CONFIGURATION_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.API_CONFIGURATION_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("API_CONFIGURATION_RESPONSE_CMD", API_CONFIGURATION_RESPONSE_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_GUI_CONFIGURATION_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_GUI_CONFIGURATION_CMD", GET_GUI_CONFIGURATION_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GUI_CONFIGURATION_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GUI_CONFIGURATION_RESPONSE_CMD", GUI_CONFIGURATION_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_NEXT_INDEX_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_NEXT_INDEX_CMD", GET_NEXT_INDEX_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.NEXT_INDEX_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("NEXT_INDEX_RESPONSE_CMD", NEXT_INDEX_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_SUPERVISOR_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_SUPERVISOR_CMD", GET_SUPERVISOR_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SUPERVISOR_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SUPERVISOR_RESPONSE_CMD", SUPERVISOR_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_DESTINATION_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_DESTINATION_CMD", CREATE_DESTINATION_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_DESTINATION_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_DESTINATION_RESPONSE_CMD", CREATE_DESTINATION_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELE

TE_DESTINATION_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_DESTINATION_CMD", DELETE_DESTINATION_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_DESTINATION_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_DESTINATION_RESPONSE_CMD", DELETE_DESTINATION_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DESTINATIONS_CMD", GET_DESTINATIONS_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GUI_CONFIGURATION_RESPONSE_CMD", GET_DESTINATIONS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_BY_ID_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DESTINATIONS_BY_ID_CMD", GET_DESTINATIONS_BY_ID_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DESTINATIONS_BY_ID_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DESTINATIONS_BY_ID_RESPONSE_CMD", GET_DESTINATIONS_BY_ID_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DESTINATION_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_DESTINATION_CMD", MODIFY_DESTINATION_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DESTINATION_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_DESTINATION_RESPONSE_CMD", MODIFY_DESTINATION_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.VALIDATE_DESTINATION_CMD_ENUM - Initial value: new

DDSXML_CommandsEnum("VALIDATE_DESTINATION_CMD", VALIDATE_DESTINATION_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.VALIDATE_DESTINATION_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("VALIDATE_DESTINATION_RESPONSE_CMD", VALIDATE_DESTINATION_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DATA_PRODUCTS_CMD", GET_DATA_PRODUCTS_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DATA_PRODUCTS_RESPONSE_CMD", GET_DATA_PRODUCTS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_BY_ID_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DATA_PRODUCTS_BY_ID_CMD", GET_DATA_PRODUCTS_BY_ID_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_PRODUCTS_BY_ID_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DATA_PRODUCTS_BY_ID_RESPONSE_CMD", GET_DATA_PRODUCTS_BY_ID_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_REQUEST_CMD", CREATE_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_REQUEST_RESPONSE_CMD", CREATE_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_REQUEST_CMD", DELETE_REQUEST_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_REQUEST_RESPONSE_CMD", DELETE_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_REQUEST_CMD", GET_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_REQUEST_RESPONSE_CMD", GET_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_REQUEST_CMD", MODIFY_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_REQUEST_RESPONSE_CMD", MODIFY_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RESUME_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("RESUME_REQUEST_CMD", RESUME_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RESUME_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("RESUME_REQUEST_RESPONSE_CMD", RESUME_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SUSPEND_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SUSPEND_REQUEST_CMD", SUSPEND_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SUSP

END_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SUSPEND_REQUEST_RESPONSE_CMD", SUSPEND_REQUEST_RESPONSE_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("TRANSFER_REQUEST_CMD", TRANSFER_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("TRANSFER_REQUEST_RESPONSE_CMD", TRANSFER_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("UPDATE_REQUEST_CMD", UPDATE_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_TEMPLATE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_TEMPLATE_CMD", CREATE_TEMPLATE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_TEMPLATE_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_TEMPLATE_RESPONSE_CMD", CREATE_TEMPLATE_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_TEMPLATE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_TEMPLATE_CMD", DELETE_TEMPLATE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_TEMPLATE_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_TEMPLATE_RESPONSE_CMD", DELETE_TEMPLATE_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_TEMPLATE_CMD_ENUM - Initial value: new

DDSXML_CommandsEnum("MODIFY_TEMPLATE_CMD", MODIFY_TEMPLATE_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_TEMPLATE_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_TEMPLATE_RESPONSE_CMD", MODIFY_TEMPLATE_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.IDLE_HANDLER_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("IDLE_HANDLER_CMD", IDLE_HANDLER_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.IDLE_HANDLER_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("IDLE_HANDLER_RESPONSE_CMD", IDLE_HANDLER_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RUN_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("RUN_REQUEST_CMD", RUN_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.RUN_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("RUN_REQUEST_RESPONSE_CMD", RUN_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DATA_SHIPMENT_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_DATA_SHIPMENT_CMD", SEND_DATA_SHIPMENT_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DATA_SHIPMENT_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_DATA_SHIPMENT_RESPONSE_CMD", SEND_DATA_SHIPMENT_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_PROGRESS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_PROGRESS_CMD", SEND_PROGRESS_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_PROGRESS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_PROGRESS_RESPONSE_CMD", SEND_PROGRESS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_STATUS_MESSAGE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_STATUS_MESSAGE_CMD", SEND_STATUS_MESSAGE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_STATUS_MESSAGE_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_STATUS_MESSAGE_RESPONSE_CMD", SEND_STATUS_MESSAGE_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.STOP_FILE_TRANSFER__CMD_ENUM - Initial value: new DDSXML_CommandsEnum("STOP_FILE_TRANSFER__CMD", STOP_FILE_TRANSFER__CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.STOP_FILE_TRANSFER_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("STOP_FILE_TRANSFER_RESPONSE_CMD", STOP_FILE_TRANSFER_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TERMINATE_HANDLER_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("TERMINATE_HANDLER_CMD", TERMINATE_HANDLER_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TERMINATE_HANDLER_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("TERMINATE_HANDLER_RESPONSE_CMD", TERMINATE_HANDLER_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRANSFER_FILE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("TRANSFER_FILE_CMD", TRANSFER_FILE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.TRAN

SFER_FILE_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("TRANSFER_FILE_RESPONSE_CMD", TRANSFER_FILE_RESPONSE_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DATA_SHIPMENT_LIST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DATA_SHIPMENT_LIST_CMD", GET_DATA_SHIPMENT_LIST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DATA_SHIPMENT_LIST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DATA_SHIPMENT_LIST_RESPONSE_CMD", DATA_SHIPMENT_LIST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_CDDR_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_CDDR_REQUEST_CMD", CREATE_CDDR_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.CREATE_CDDR_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("CREATE_CDDR_REQUEST_RESPONSE_CMD", CREATE_CDDR_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_CDDR_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_CDDR_REQUEST_CMD", DELETE_CDDR_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_CDDR_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_CDDR_REQUEST_RESPONSE_CMD", DELETE_CDDR_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_CDDR_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_CDDR_REQUEST_CMD", GET_CDDR_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_CDDR_REQUEST_RESPONSE_CMD_ENUM - Initial value: new

DDSXML_CommandsEnum("GET_CDDR_REQUEST_RESPONSE_CMD", GET_CDDR_REQUEST_RESPONSE_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_CDDR_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_CDDR_REQUEST_CMD", MODIFY_CDDR_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_CDDR_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_CDDR_REQUEST_RESPONSE_CMD", MODIFY_CDDR_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DENY_DATA_LIST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DENY_DATA_LIST_CMD", GET_DENY_DATA_LIST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_DENY_DATA_LIST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_DENY_DATA_LIST_RESPONSE_CMD", GET_DENY_DATA_LIST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_HANDLER_ERROR_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_HANDLER_ERROR_CMD", SEND_HANDLER_ERROR_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_HANDLER_STATUS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_HANDLER_STATUS_CMD", SEND_HANDLER_STATUS_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_HANDLER_STATUS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_HANDLER_STATUS_RESPONSE_CMD", SEND_HANDLER_STATUS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_REQUESTS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_ALL_REQUESTS_CMD", DELETE_ALL_REQUESTS_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_REQUESTS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_ALL_REQUESTS_RESPONSE_CMD", DELETE_ALL_REQUESTS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_DESTINATION_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("UPDATE_DESTINATION_CMD", UPDATE_DESTINATION_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_SUPERVISOR_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("UPDATE_SUPERVISOR_CMD", UPDATE_SUPERVISOR_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_BY_ID_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_REQUEST_BY_ID_CMD", GET_REQUEST_BY_ID_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_BY_ID_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_REQUEST_BY_ID_RESPONSE_CMD", GET_REQUEST_BY_ID_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_RETRANSMIT_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("JMS_RETRANSMIT_REQUEST_CMD", JMS_RETRANSMIT_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_RETRANSMIT_REQUEST_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("JMS_RETRANSMIT_REQUEST_RESPONSE_CMD", JMS_RETRANSMIT_REQUEST_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_REQUEST_IDS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_REQUEST_IDS_CMD", GET_REQUEST_IDS_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_

REQUEST_IDS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_REQUEST_IDS_RESPONSE_CMD", GET_REQUEST_IDS_RESPONSE_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_CDDR_REQUESTS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("DELETE_ALL_REQUESTS_CMD", DELETE_ALL_REQUESTS_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.DELETE_ALL_CDDR_REQUESTS_RESPONSE_CMD_ENUM -Initial value: new DDSXML_CommandsEnum("DELETE_ALL_REQUESTS_RESPONSE_CMD", DELETE_ALL_REQUESTS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_CDDR_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("UPDATE_CDDR_REQUEST_CMD", UPDATE_CDDR_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_NEW_REQUEST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_NEW_REQUEST_CMD", GET_NEW_REQUEST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DENY_DATA_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_DENY_DATA_CMD", MODIFY_DENY_DATA_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.MODIFY_DENY_DATA_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("MODIFY_DENY_DATA_RESPONSE_CMD", MODIFY_DENY_DATA_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.UPDATE_DENY_DATA_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("UPDATE_DENY_DATA_CMD", UPDATE_DENY_DATA_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_MASTER_DESTINATIONS_CMD_ENUM - Initial value: new

DDSXML_CommandsEnum("GET_MASTER_DESTINATIONS_CMD", GET_MASTER_DESTINATIONS_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_MASTER_DESTINATIONS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_MASTER_DESTINATIONS_RESPONSE_CMD", GET_MASTER_DESTINATIONS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_FILTERED_DP_LIST_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_FILTERED_DP_LIST_CMD", GET_FILTERED_DP_LIST_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.GET_SYSTEM_MESSAGES_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("GET_SYSTEM_MESSAGES_CMD", GET_SYSTEM_MESSAGES_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DENY_DATA_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_DENY_DATA_CMD", SEND_DENY_DATA_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DENY_DATA_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("SEND_DENY_DATA_RESPONSE_CMD", SEND_DENY_DATA_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_SUBSCRIBE_TO_STATUS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("JMS_SUBSCRIBE_TO_STATUS_CMD", JMS_SUBSCRIBE_TO_STATUS_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_SUBSCRIBE_TO_STATUS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("JMS_SUBSCRIBE_TO_STATUS_RESPONSE_CMD", JMS_SUBSCRIBE_TO_STATUS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_STOP_SUBSCRIBE_TO_STATUS_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("JMS_STOP_SUBSCRIBE_TO_STATUS_CMD", JMS_STOP_SUBSCRIBE_TO_STATUS_CMD)

- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.JMS_STOP_SUBSCRIBE_TO_STATUS_RESPONSE_CMD_ENUM - Initial value: new DDSXML_CommandsEnum("JMS_STOP_SUBSCRIBE_TO_STATUS_RESPONSE_CMD", JMS_STOP_SUBSCRIBE_TO_STATUS_RESPONSE_CMD)
- static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.SEND_DDR_CMD - Initial value: new DDSXML_CommandsEnum("SEND_DDR_CMD", SEND_DDR_CMD)

3.3.37.3 DDSXML_CommandsEnum Functions

3.3.37.3.1 DDSXML_CommandsEnum.findByName

static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_CMD_ENUM

Parameters:

- val The string representing an enums name

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

3.3.37.3.2 DDSXML_CommandsEnum.findByValue

static DDSXML_CommandsEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandsEnum.findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_CMD_ENUM

Parameters:

- val The int representing an enums value

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

3.3.38 DDSXML_CommandStateEnum Class Reference

This class is defines all command states

The Class diagram representing the DDSXML_CommandStateEnum Class is provided in Figure 3.3.38-1, DDSXML_CommandStateEnum Class UML Diagram.

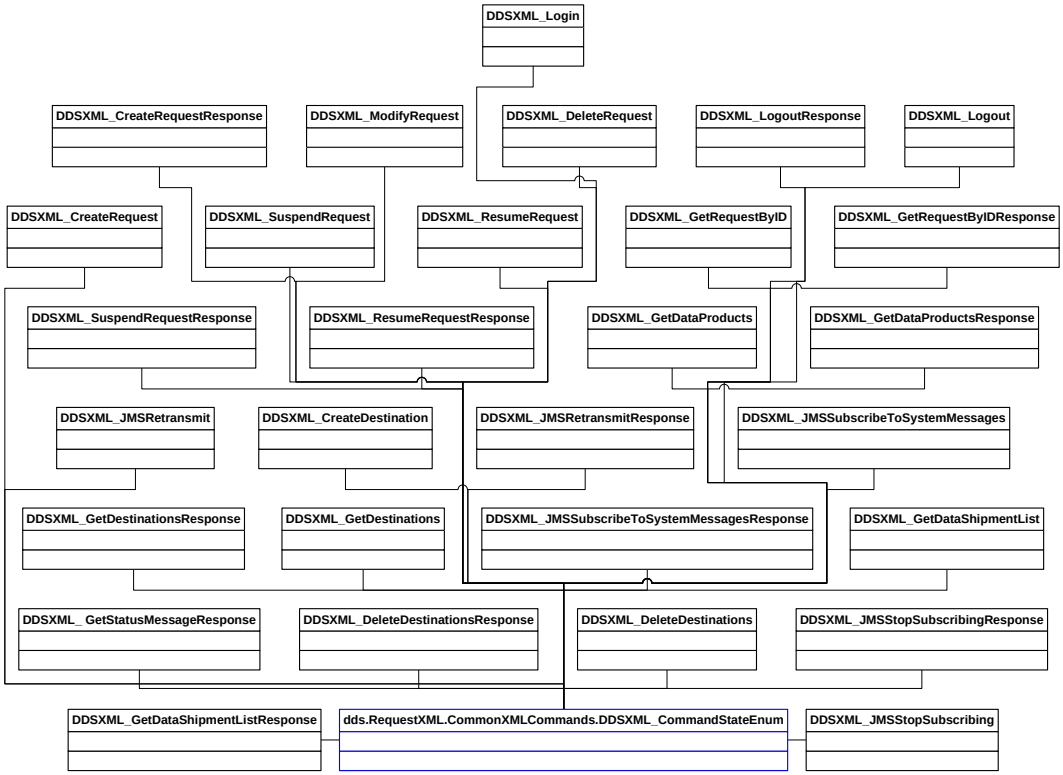

Figure 3.3.38-1, DDSXML_CommandStateEnum Class UML Diagram

3.3.38.1 DDSXML_CommandStateEnum Attributes

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.CMD_STATE_UNKNOWN = 0
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.CMD_STATE_SUCCESSFUL = 1
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.CMD_STATE_FAILURE = 2
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.MIN_VALUE = CMD_STATE_UNKNOWN
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.MAX_VALUE = CMD_STATE_FAILURE + 1

3.3.38.2 DDSXML_CommandStateEnum Enumerations

- static DDSXML_CommandStateEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.CMD_STATE_UNKNOWN_ENUM - Initial value: new DDSXML_CommandStateEnum("CMD_STATE_UNKNOWN", CMD_STATE_UNKNOWN)
- static DDSXML_CommandStateEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.CMD_STATE_SUCCESSFUL_ENUM - Initial value: new DDSXML_CommandStateEnum("CMD_STATE_SUCCESSFUL", CMD_STATE_SUCCESSFUL)
- static DDSXML_CommandStateEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.CMD_STATE_FAILURE_ENUM - Initial value: new DDSXML_CommandStateEnum("CMD_STATE_FAILURE", CMD_STATE_FAILURE)

3.3.38.3 DDSXML_CommandStateEnum Functions

3.3.38.3.1 DDSXML_CommandStateEnum.findByName

static DDSXML_CommandStateEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.findByName (

String name

)

Find the enum that corresponds to the given string. If can't find it just return CMD_STATE_UNKNOWN

Parameters:

- val The string representing an enums name

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

3.3.38.3.2 DDSXML_CommandStateEnum.findByValue

static DDSXML_CommandStateEnum dds.RequestXML.CommonXMLCommands.DDSXML_CommandStateEnum.findByValue (

int value

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_CMD

Parameters:

- val The int representing an enums value

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

3.3.39 DDSXML_ClientTypesEnum Class Reference

This class is defines all command states This must match the C++

The Class diagram representing the DDSXML_JMSStopSubscribingResponse Class is provided in Figure 3.3.39-1, DDSXML_JMSStopSubscribingResponse Class UML Diagram.

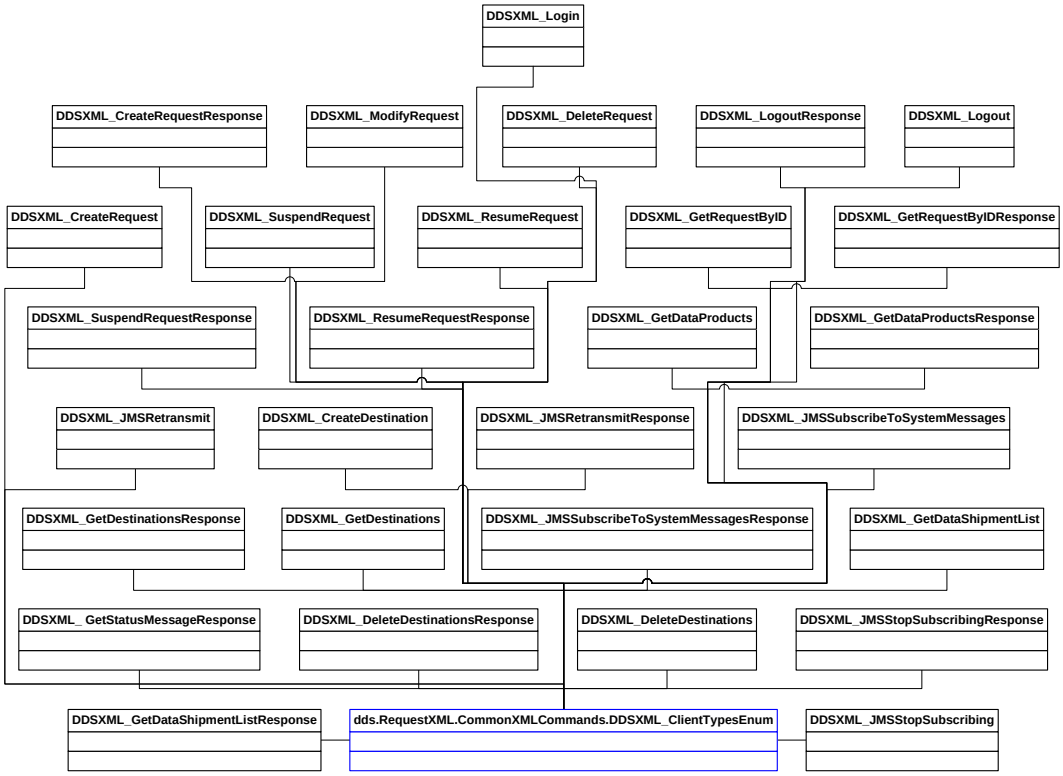

Figure 3.3.39-1, DDSXML_ClientTypesEnum Class UML Diagram

3.3.39.1 DDSXML_ClientTypesEnum Attributes

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.CLIENT_API = 0
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.GUI_API = 1

- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.HANDLER_API = 2
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.FILE_TRANSFER_API = 3
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.SERVER_API = 4
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.UNKNOWN_API = 5
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.MIN_VALUE = CLIENT_API
- static final int dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.MAX_VALUE = UNKNOWN_API + 1

3.3.39.2 DDSXML_ClientTypesEnum Enumerations

- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.CLIENT_API_ENUM - Initial value: new DDSXML_ClientTypesEnum("CLIENT_API", CLIENT_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.GUI_API_ENUM - Initial value: new DDSXML_ClientTypesEnum("GUI_API", GUI_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.HANDLER_API_ENUM - Initial value: new DDSXML_ClientTypesEnum("HANDLER_API", HANDLER_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.FILE_TRANSFER_API_ENUM - Initial value: new DDSXML_ClientTypesEnum("FILE_TRANSFER_API", FILE_TRANSFER_API)
- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.SERVER_API_ENUM - Initial value: new DDSXML_ClientTypesEnum("SERVER_API", SERVER_API)

- static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.UNKNOWN_API_ENUM - Initial value: new DDSXML_ClientTypesEnum("UNKNOWN_API", UNKNOWN_API)

3.3.39.3 DDSXML_ClientTypesEnum Functions

3.3.39.3.1 DDSXML_ClientTypesEnum.findByName

static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.findByName(

String name

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_API

Parameters:

- val The string representing an enums name

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

3.3.39.3.2 DDSXML_ClientTypesEnum.findByValue

static DDSXML_ClientTypesEnum dds.RequestXML.CommonXMLCommands.DDSXML_ClientTypesEnum.findByValue(

int value

)

Find the enum that corresponds to the given string. If can't find it just return UNKNOWN_CMD

Parameters:

- val The int representing an enums value

Returns:

- The enum object if found else UNKNOWN_DATA_TYPE

3.3.40 DDSXML_User Class Reference

See Java API Documentation for the signature for the DDSXML_User Class Reference.

3.3.41 DDSXML_Data Class Reference

This base class is the base for all data in the DDS system. This class is based on the Base XML class. All pointers returned, referenced memory is owned by the API and should not be destroyed by the caller unless noted in the method called.

The Class diagram representing the DDSXML_JMSStopSubscribingResponse Class is provided in Figure 3.3.41-1, DDSXML_JMSStopSubscribingResponse Class UML Diagram.

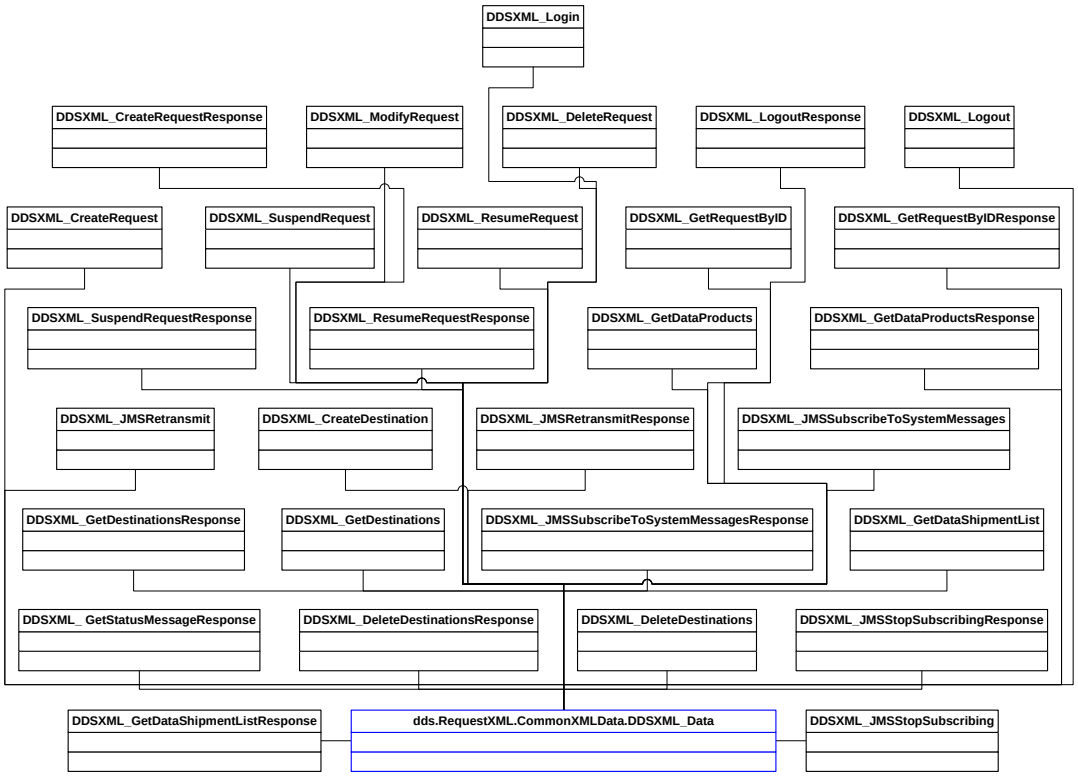

Figure 3.3.41-1, DDSXML_Data Class UML Diagram

3.3.41.1 DDSXML_Data Attributes

- protected DDSXML_BaseStatusHandler dds.RequestXML.CommonXMLData.DDSXML_Data.myStatusHandler = null - Status Handler This is the handler for status messages.

3.3.41.2 DDSXML_Data Enumerations

- protected DDSXML_DataStateEnum dds.RequestXML.CommonXMLData.DDSXML_Data.myDataState = DDSXML_DataStateEnum.DATA_STATE_UNKNOWN_ENUM - The State for this data item

3.3.41.3 DDSXML_Data Functions

3.3.41.3.1 DDSXML_Data.equals

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.equals (

DDSXML_Data data

)

Compare operator==

Parameters:

- user The DDSXML_User that the data is to be compared to.

Returns:

- boolean true if the two are equal.

3.3.41.3.2 DDSXML_Data.compareTo

int dds.RequestXML.CommonXMLData.DDSXML_Data.compareTo (

Object arg0

)

The function is required by the comparable interface, and is used to determine the order of te objects

Parameters:

- arg0 The object to compare against.

Returns:

- Return -1 if less, 0 if same, 1 if greater

3.3.41.3.3 DDSXML_Data.lessThan

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.lessThan (

DDSXML_Data data

)

Compare lessThan

Parameters:

- user The DDSXML_User that the data is to be compared to.

Returns:

- boolean true if <.

3.3.41.3.4 DDSXML_Data.lessThanOrEquals

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.lessThanOrEquals (

DDSXML_Data data

)

Compare lessThanOrEquals

Parameters:

- user The DDSXML_User that the data is to be compared to.

Returns:

- boolean true if <=.

3.3.41.3.5 DDSXML_Data.greaterThan

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.greaterThan (

DDSXML_Data data

)

Compare greaterThan

Parameters:

- user The DDSXML_User that the data is to be compared to.

Returns:

- boolean true if >.

3.3.41.3.6 DDSXML_Data.greaterThanOrEquals

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.greaterThanOrEquals (

DDSXML_Data data

)

Compare operator>=

Parameters:

- user The DDSXML_User that the data is to be compared to.

Returns:

- boolean true if >=.

Exceptions:

- DDSAPI_XMLException if initialization fails, or a parse error is encountered.

3.3.41.3.7 DDSXML_Data.validate

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.validate (

)

This method validates the data.

Returns:

- boolean true if the data is valid.

3.3.41.3.8 DDSXML_Data.validateData

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.validateData (

boolean IDFlag

)

This method validates the XML.

Parameters:

- IDFlag If the flag is set validate this field

Returns:

- boolean true if the data is valid.

3.3.41.3.9 DDSXML_Data.getDataTypeName

String dds.RequestXML.CommonXMLData.DDSXML_Data.getDataTypeName (

)

This is the data type of this data object as a string. All DDS data items have a data type.

Returns:

- std:string - The data type of this object.

3.3.41.3.10 DDSXML_Data.getDataType

DDSXML_DataTypesEnum
dds.RequestXML.CommonXMLData.DDSXML_Data.getDataType (

)

This is the data type of this data object. All DDS data items have a data type.

Returns:

- DDSXML_DataTypes - The data type of this object.

3.3.41.3.11 DDSXML_Data.getName

String dds.RequestXML.CommonXMLData.DDSXML_Data.getName (

)

This method gets the Name of this data item. This is a name the user makes up.

Returns:

- name The name for this data.

3.3.41.3.12 DDSXML_Data.setName

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.setName (

String name

)

This method sets the Name of this data item. This is a name the user makes up.

Parameters:

- name The name for this data.

Returns:

- boolean - true if successful.

3.3.41.3.13 DDSXML_Data.getModifyEnabled

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.getModifyEnabled (

)

This lets the user know if this data can be modified. This can be used by derived classes to let the caller know that the data is not able to be modified.

Returns:

- boolean - true if this data can be modified.

3.3.41.3.14 DDSXML_Data.getDataStateName

String dds.RequestXML.CommonXMLData.DDSXML_Data.getDataStateName (

)

This is the data state of this data object as a string. All DDS data items have a data state.

Returns:

- std:string - The data state of this object.

3.3.41.3.15 DDSXML_Data.getDataState

DDSXML_DataStateEnum
dds.RequestXML.CommonXMLData.DDSXML_Data.getDataState (

)

This is the data state of this data object. All DDS data items have a data state.

Returns:

- DDSXML_DataTypes - The data type of this object.

3.3.41.3.16 DDSXML_Data.setDataState

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.setDataState (

DDSXML_DataStateEnum datastate

)

This is the data state of this data object. All DDS data items have a data state.

Parameters:

- std:string - The data type of this object.

Returns:

- boolean - true if successful.

3.3.41.3.17 DDSXML_Data.setStatusHandler

DDSXML_BaseStatusHandler
dds.RequestXML.CommonXMLData.DDSXML_Data.setStatusHandler (

DDSXML_BaseStatusHandler statusHandler

)

Sets the statusHandler for this request to send status to. We own the object passed in.

Parameters:

- statusHandler The status Handler

Returns:

- DDSXML_BaseStatusHandler - Original DDSXML_BaseStatusHandler or null if none.

3.3.41.3.18 DDSXML_Data.setDataType

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.setDataType (

DDSXML_DataTypesEnum datatype

)

This is the data type of this data object. All DDS data items have a data type.

Parameters:

- std:string - The data type of this object.

Returns:

- boolean - true if successful.

3.3.41.3.19 DDSXML_Data.setModifyEnabled

boolean dds.RequestXML.CommonXMLData.DDSXML_Data.setModifyEnabled (

boolean modifyFlag

)

This lets the user know if this data can be modified. This can be used by derived classes to let the caller know that the data is not able to be modified.

Parameters:

- boolean - true if this data can be modified.

Returns:

- boolean - true if this data can be modified.

3.3.41.3.20 DDSXML_Data.getCreatedTimestamp

package long
dds.RequestXML.CommonXMLData.DDSXML_Data.getCreatedTimestamp (

)

This method gets the time this object was created.

Returns:

- long time this object was created.

3.3.41.3.21 DDSXML_Data.setCreatedTimestamp

package boolean
dds.RequestXML.CommonXMLData.DDSXML_Data.setCreatedTimestamp (

long ietTime

)

This method sets the time this object was created.

Parameters:

- long time this object was created.

Returns:

- boolean - true if successful.

3.3.41.3.22 DDSXML_Data.getLastModifiedTimestamp

package long
dds.RequestXML.CommonXMLData.DDSXML_Data.getLastModifiedTimestamp (

)

This method gets the time this object was last modified.

Returns:

- time_t time this object was last modified.

3.3.41.3.23 DDSXML_Data.setLastModifiedTimestamp

package boolean
dds.RequestXML.CommonXMLData.DDSXML_Data.setLastModifiedTimestamp (

long ietTime

)

This method sets the time this object was last modified.

Parameters:

- long time this object was modified.

Returns:

- boolean - true if successful.

APPENDIX A SYSTEM REQUIREMENTS

The Installation Guide for the NPOESS API is documented in the Installation Guide which accompanies the software as delivered by NPOESS. The Installation Guide file is identified in the distribution by the filename: INSTALL-JAVA, INSTALL-CPP, INSTALL-JMS. See the appropriate file details.

The NPOESS API is designed for use on the Microsoft® Windows® 2000 platform and on IBM AIX® (Advanced IBM Unix) operating systems. See the NPOESS API User's Guide for the Commercial Off-The-Shelf (COTS) for a complete listing of the products tested with the NPOESS API.

For the Java and JMS APIs the following compilers are used:

- Java API
  - Version 1.4 of the Java Development Kit (JDK)
- Commercial Off-The-Shelf (COTS) Products:
  - IBM AIX®
    - Version 2.6.2 of Xerces-J[1] (Java)
    - Version 7.0 of Borland® VisiBroker®
    - Version 1.4 of Apache Axis
    - Version 1.2.8 of Apache Jakarta-Log4j
    - Version 1.0.4 of Apache Jakarta Commons Logging
    - Version 1.4.2 SR11 of Java SDK
    - Version 1.4.2.SR11 of Java SDK 64bit
    - Version 1.1 Java JMS
  - Microsoft® Windows®
    - Version 2.6.2 of Xerces (Java)
    - Version 7.0 of Borland® VisiBroker®
    - Version 1.4 of Apache Axis
    - Version 1.2.8 of Apache Jakarta-Log4j
    - Version 1.0.4 of Apache Jakarta Commons Logging
    - Version 1.4.2 SR11 of Java SDK

[1] Xerces (named after the Xerces Blue butterfly) provides XML parsing and generation for both Java and C++.

APPENDIX B DOCUMENT SPECIFIC ACRONYMS LIST

This table identifies and defines acronyms unique to this document. All other acronyms are listed and identified in the NPOESS Program Acronyms, D35838.

Table B-1, Document-Specific Acronym List

Acronym	Definition
AIX	Advanced IBM Unix
IETF	Internet Engineering Task Force
JDK	Java Development Kit
JVM	Java Virtual Machine
SEITO	System Engineering, Integration, Test and Operations
W3C	World Wide Web Consortium