

Directorate of Operations and Infrastructure

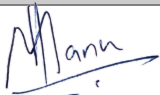
EXTERNAL USER INTERFACE CONTROL DOCUMENT (EUICD)

EGOS Data Dissemination
System (EDDS)

Reference: EGOS-GEN-EDDS-ICD-1001
Version: 24.0
Date: 2025-01-10

Document Title:	External User Interface Control Document (EUICD)		
Document Reference:	EGOS-GEN-EDDS-ICD-1001		
Document Version:	24.0	Date:	2025-01-10
Abstract			

Approval Table:

Action	Name	Function	Signature	Date
Prepared by:	EDDS Team	COSMIC		2025-01-10
Verified by:	Florian Cleysac	Quality Assurance Manager		2025-01-10
Approved by:	Rui Santos	Technical Officer		2025-01-10

Authors and Contributors:

Name	Contact	Description	Date
M. Hawkshaw	michael.hawkshaw@cgi.com	Author	2016-10-19
R. Ots	rauno.ots@cgi.com	Contributor	2016-10-19
K. Panitzek	kamill.panitzek@cgi.com	Contributor	2016-10-19
Rokibul Uddin	rokibul.uddin@c-ssystems.de	Contributor	2017-05-05 2018-02-16
Abhishek Annadanappa Chandrashekar	abhishek.annadanappa@c-ssystems.de	Contributor	2020-09-30
Teresa Noviello	teresa.noviello@c-ssystems.de	Contributor	2021-05-31
AlaEddine Zaghdoud	alaeddine.zaghdoud@c-ssystems.de	Contributor	2022-10-31 2023-09-08
Manu Mohan	manu.mohan@csgroup.de	Contributor	2023-02-20 2023-04-06 2023-11-01 2024-02-28 2024-05-07

Distribution List:

© COPYRIGHT EUROPEAN SPACE AGENCY, 2025

The copyright of this document is vested in the European Space Agency. This document may only be reproduced in whole or in part, stored in a retrieval system, transmitted in any form, or by any means e.g. electronically, mechanically or by photocopying, or otherwise, with the prior permission of the Agency.

Document Change Log

Issue	Date	Description
1.0	2007-04-13	Update in line with PDR outcome.
2.0	2010-10-11	Issue of document for PA delivery
2.1	2011-03-29	Issue for Final Acceptance Delivery

Issue	Date	Description
3.0	2011-09-09	Issue for EDDS v1.1.0i1
3.1	2011-10-24	Issue for EDDS v1.1.0i2
3.2	2011-11-14	Issue for EDDS v1.1.0i3
3.3	2012-02-29	Issue for EDDS v1.1.1i1
3.4	2012-04-04	Issue for EDDS v1.1.1i2
3.5	2012-05-18	Issue for EDDS v1.1.2i1
3.6	2012-09-17	Issue for EDDS v1.2.0i1
3.7	2012-09-27	Issue for EDDS v1.2.0i2
3.8	2012-10-12	Issue for EDDS v1.2.0i3
3.9	2013-03-21	Issue for EDDS v1.2.1i1
3.10	2013-04-10	Issue for EDDS v1.2.1i2
4.0	2013-06-21	Issue for EDDS v1.2.2i1
5.0	2013-12-03	Issue for EDDS v1.3.0i1
6.0	2014-05-28	Issue for EDDS v1.4.0i1
7.0	2014-12-17	Issue for EDDS v1.5.0i1
8.0	2016-02-04	Issue for EDDS v1.6.0i1
9.0	2016-07-04	Issue for EDDS v.2.0.0i1
11.0	2016-10-19	Issue for EDDS v.2.1.0i1
13.0	2017-05-05	Issue for EDDS v.2.2.0i1
15.0	2018-01-15	Issue for EDDS v2.3.0i1
16.0	2020-02-11	Issue for EDDS v2.4.0i1
17.0	2020-09-30	Issue for EDDS v3.0.0i1
18.0	2021-05-31	Issue for EDDS v3.1.0i1
19.0	2022-09-30	Issue for EDDS v3.2.0i1
20.0	2022-09-30	Issue for EDDS v3.3.0i1
21.0	2023-02-09	Issue for EDDS v3.3.1i1
22.0	2023-11-03	Issue for EDDS v3.4.0i1
23.0	2024-03-08	Issue for EDDS v3.5.0i1
24.0	2025-01-10	Issue for EDDS v3.6.0i1

Document Change Record

DCR No:	01		
Date:	2024-05-07		
Document Title:	External User Interface Control Document (EUICD)		
Document Reference:	EGOS-GEN-EDDS-ICD-1001		
Page	Paragraph	Reason for Change	
217,218,219,220 256, 259, 260, 261,262,263,264,265, 266,267	D.45, D.46, D.47, D.48, D.49 D.50, D.51, D.52, G.69, G.70, G.71, G.74, G.76, G.77, G.78, G.79, G.80, G.81, G.82, G.83, G.84, G.85, G.86, G.87	EDDS-1355 : Prefiltering of EGSCC requests in EDDS	

DCR No:	02		
Date:	2023-08-26		
Document Title:	External User Interface Control Document (EUICD)		
Document Reference:	EGOS-GEN-EDDS-ICD-1001		
Page	Paragraph	Reason for Change	
262, 286, 52, 50	G.70, I.14, I.18, 6.7.3.3.8, 6.7.3.3.10	EDDS-1373 : Align Kafka and EDDS response fields	

DCR No:	03		
Date:	2023-08-26		
Document Title:	External User Interface Control Document (EUICD)		
Document Reference:	EGOS-GEN-EDDS-ICD-1001		
Page	Paragraph	Reason for Change	
57, 58, 59	6.7.3.3.13.1	EDDS-1367 : Activity Argument Data mapped to TM Parameter proto	

DCR No:	04		
Date:	2023-09-26		
Document Title:	External User Interface Control Document (EUICD)		
Document Reference:	EGOS-GEN-EDDS-ICD-1001		
Page	Paragraph	Reason for Change	
61, 62, 63, 64, 65, 66, 210, 277, 311	6.7.3.3.16, C.100, G.88, G.89 , G.90, I.50	EDDS-1271 : Add Egsccl OOL requests to EDDS	

DCR No:	05		
Date:	2024-11-20		
Document Title:	External User Interface Control Document (EUICD)		
Document Reference:	EGOS-GEN-EDDS-ICD-1001		
Page	Paragraph	Reason for Change	
393, 394	K.19, K.20	EDDS-1376 : Extend EDDS quotas to support restriction for TC requests	

TABLE OF CONTENTS

1. INTRODUCTION..... 1

1.1 PURPOSE..... 1

1.2 SCOPE..... 1

1.3 DOCUMENT OVERVIEW..... 1

2. REFERENCES..... 3

2.1 APPLICABLE DOCUMENTS 3

2.2 REFERENCE DOCUMENTS 3

3. GLOSSARY 4

3.1 ACRONYMS..... 4

3.2 DEFINITION OF TERMS..... 5

4. SOFTWARE OVERVIEW 6

4.1 EDDS SERVICES OVERVIEW 6

4.1.1 Authentication Services 6

4.1.2 Batch Service..... 6

4.1.3 Stream Service..... 6

4.1.4 User Management Service..... 9

4.2 EDDS INTERFACE OVERVIEW 9

4.2.1 EDDS Web Service Interface 9

4.2.2 Stream Web Service Interface 9

4.2.3 Email Interface 9

4.2.4 Delivery Manager Interface 9

4.3 CLIENT APPLICATIONS OVERVIEW 9

4.3.1 EDDS Client Application 10

4.3.2 Email Server 10

4.3.3 Delivery Manager..... 11

4.3.4 EDDS Stream client..... 11

4.3.5 Request Submitter 11

5. DATA TYPES..... 17

5.1 PACKET 17

5.2 PARAMETER..... 17

5.3 REPORT 18

5.4 ARCHIVED FILES..... 18

5.5 ACKNOWLEDGEMENT 19

5.6 REQUEST AND RESPONSE MAPPINGS..... 19

6. FORMATS 21

6.1 EDDS BINARY..... 23

6.1.1 EDDS Raw Header 23

6.1.2 Packet Data..... 24

6.1.3 GDDS Binary File Example..... 30

6.2 XML 33

6.2.1 Packet..... 33

6.2.2 Parameter 33

6.2.3 Report..... 34

6.2.4 Archived Files 35

6.2.5 File System 35

6.2.6 Acknowledgement 35

6.3 XML TRANSFORM 35

6.4 XFDU 35

6.5 TDRS SPREADSHEET 35

6.6 ASCII..... 35

6.7 BINARY 37

6.7.1 Archive File 37

6.7.2	File System	37
6.7.3	Google Proto Buffers	38
6.8	RAWSOURCEBINARY	66

7. THE OUTPUT IS A BINARY FILE OF THE CONTINUOUS RAW DATA AS PROVIDED BY RAWBODYDATA ATTRIBUTE. NAMING CONVENTIONS 67

7.1	REQUEST ID CONVENTION.....	67
7.2	RESPONSE ID CONVENTION.....	67
7.2.1	EDDS Stream Client’s response file name convention.....	68

8. DELIVERY 69

8.1	DELIVERY MECHANISM	69
8.1.1	File Server.....	69
8.1.2	EDDS Server	69
8.1.3	Stream	69

9. WEB SERVICES DESCRIPTION 70

9.1	WEB SERVICE INTRODUCTION	70
9.2	PORTTYPES.....	70
9.2.1	EDDS Port Type	71
9.3	MESSAGES	87
9.3.1	message accountRequest.....	87
9.3.2	message accountRequestResponse.....	87
9.3.3	message authenticationFault.....	87
9.3.4	message authorizationFault.....	88
9.3.5	message batchRequest.....	88
9.3.6	message batchRequestResponse.....	89
9.3.7	message cancel	89
9.3.8	message suspend	89
9.3.9	message resume.....	90
9.3.10	message cancelResponse	90
9.3.11	message suspendResponse.....	90
9.3.12	message resumeResponse.....	90
9.3.13	message checkSession	91
9.3.14	message checkSessionResponse.....	91
9.3.15	message checkUserPassword	91
9.3.16	message checkUserPasswordResponse	91
9.3.17	message deleteData	92
9.3.18	message deleteDataResponse	92
9.3.19	message deleteRequest.....	92
9.3.20	message deleteRequestResponse.....	93
9.3.21	message getAllowedRequestTypes.....	93
9.3.22	message getAllowedRequestTypesResponse.....	93
9.3.23	message getCatalogue.....	93
9.3.24	message getCatalogueResponse.....	94
9.3.25	message getDataspaces	94
9.3.26	message getDataspacesResponse	94
9.3.27	message getFarcCatalogue.....	94
9.3.28	message getFarcCatalogueResponse.....	95
9.3.29	message getHistoricalLog.....	95
9.3.30	message getHistoricalLogResponse	95
9.3.31	message getJobs	96
9.3.32	message getJobsResponse	96
9.3.33	message getJobsWithStatus	96
9.3.34	message getJobsWithStatusAndLimit.....	96
9.3.35	message getJobsWithStatusAndLimitResponse.....	97
9.3.36	message getJobsWithStatusResponse	97
9.3.37	message getLastConsolidation	97
9.3.38	message getLastConsolidationResponse.....	97
9.3.39	message getMissionDetails.....	98

9.3.40	message getMissionDetailsResponse.....	98
9.3.41	message getMissions.....	98
9.3.42	message getMissionsResponse.....	98
9.3.43	message getMissionsAndDomains.....	99
9.3.44	message getMissionsAndDomainsResponse.....	99
9.3.45	message getParamDefinitions.....	99
9.3.46	message getParamDefinitionsResponse.....	99
9.3.47	message getQuotaDetails.....	100
9.3.48	message getQuotaDetailsResponse.....	100
9.3.49	message getRequest.....	100
9.3.50	message getRequestResponse.....	100
9.3.51	message getResponse.....	101
9.3.52	message getResponseResponse.....	101
9.3.53	message getRoles.....	101
9.3.54	message getRolesResponse.....	101
9.3.55	message getStatus.....	102
9.3.56	message getStatuses.....	102
9.3.57	message getStatusesResponse.....	102
9.3.58	message getStatusResponse.....	103
9.3.59	message getTransformations.....	103
9.3.60	message getTransformationsResponse.....	103
9.3.61	message getUserAccountDetails.....	103
9.3.62	message getUserAccountDetailsResponse.....	104
9.3.63	message getUserQuota.....	104
9.3.64	message getUserQuotaResponse.....	104
9.3.65	message getUsers.....	104
9.3.66	message getUsersResponse.....	105
9.3.67	message hasPermissionToSeeAllUsers.....	105
9.3.68	message logIn.....	105
9.3.69	message logInResponse.....	105
9.3.70	message logOut.....	106
9.3.71	message passwordExpiredFault.....	106
9.3.72	message permissionToSeeAllUsersResponse.....	106
9.3.73	message requestFault.....	107
9.3.74	message reSubmit.....	107
9.3.75	message reSubmitResponse.....	108
9.3.76	message sessionFault.....	108
9.3.77	message streamRequest.....	109
9.3.78	message streamRequestResponse.....	109
9.3.79	message startStreamData.....	109
9.3.80	message startStreamDataResponse.....	109
9.3.81	message stopHistoricalLogRetrieval.....	110
9.3.82	message stopHistoricalLogRetrievalResponse.....	110
9.3.83	message stopStreamData.....	110
9.3.84	message getStreamRequest.....	110
9.3.85	message getStreamRequestResponse.....	111
9.3.86	message getAlias.....	111
9.3.87	message getAliasResponse.....	111
9.3.88	message checkSession.....	111
9.3.89	message stopStreamDataResponse.....	111
9.4	BINDINGS.....	112
9.5	EDDS REST API ENDPOINTS.....	120
9.5.1	REST API Authentication.....	127
9.5.2	REST API Testing.....	127
10.	E-MAIL INTERFACE.....	130
10.1	DATA ELEMENTS.....	130
10.2	MESSAGE DESCRIPTION.....	130
10.3	COMMUNICATION PROTOCOL.....	130
10.4	ERRORS.....	130

11. FILE SERVER INTERFACE..... 131

11.1 DATA ELEMENTS 131

11.2 MESSAGE DESCRIPTION 131

11.3 COMMUNICATION PROTOCOL..... 131

11.4 ERRORS 131

12. CLIENT API..... 132

12.1 EDDS WEB SERVICES CLIENT API..... 133

12.1.1 Instantiating 133

12.1.2 Session Management 133

12.1.3 Issuing Requests 133

12.1.4 Managing Requests 133

12.1.5 Special Requests..... 134

12.1.6 Utility Methods for Requests..... 134

12.1.7 Utility Methods for User Management 135

12.1.8 Edds Client API – code examples 135

12.1.9 EDDS Client API – setup a client 142

12.2 EDDS PUSH NOTIFICATIONS CLIENT 148

12.2.1 Security 148

12.2.2 Example: Register notifications listener..... 148

12.3 EDDS UTILITY METHODS 149

12.3.1 Decoding EDDS Param Binary format..... 149

12.4 DEPENDENCIES 149

APPENDIX A ACKNOWLEDGEMENT DATA TYPE 150

A.1 COMPLEXTYPE ACKNOWLEDGEMENTPART..... 150

A.2 COMPLEXTYPE ACKNOWLEDGEMENTPARTLIST 151

A.3 COMPLEXTYPE DARCRETRIEVEREFERENCE..... 151

A.4 COMPLEXTYPE FILESYSTEMNAMEREFERENCE..... 151

A.5 COMPLEXTYPE PARAMETERSAMPLEREFERENCE..... 152

A.6 COMPLEXTYPE RESPONSEFILEENTRY..... 152

A.7 COMPLEXTYPE RETRIEVALREFERENCE 153

A.8 COMPLEXTYPE SAMPLEREFERENCE..... 153

A.9 COMPLEXTYPE SAMPLETIMESTAMP 153

A.10 SIMPLETYPE RESPONSEFILESTATUS 153

APPENDIX B AUTHENTICATION DATA TYPE 155

B.1 COMPLEXTYPE USERCREDENTIALS..... 155

APPENDIX C BATCH REQUEST DATA TYPE..... 156

C.1 COMPLEXTYPE ALIASREQUEST 156

C.2 COMPLEXTYPE ALIASTYPE..... 156

C.3 COMPLEXTYPE ALLOWEDREQUESTTYPES..... 156

C.4 COMPLEXTYPE ARCHIVECATALOGUE..... 157

C.5 COMPLEXTYPE ARCHIVEFILE 158

C.6 COMPLEXTYPE ARCHIVESUBSCRIPTION 159

C.7 COMPLEXTYPE ARCHIVESUBSCRIPTIONNOTIFICATION 160

C.8 COMPLEXTYPE BATCHREQUEST..... 161

C.9 COMPLEXTYPE BEHAVIOURSTATE..... 161

C.10 COMPLEXTYPE BEHAVIOURSTATES..... 162

C.11 COMPLEXTYPE CANCELPART..... 162

C.12 COMPLEXTYPE CANCELPARTLIST..... 162

C.13 COMPLEXTYPE CATALOGUEREQUEST 163

C.14 COMPLEXTYPE CATALOGUERESPONSE..... 163

C.15 COMPLEXTYPE CATALOGUERESPONSEELEMENT..... 164

C.16 COMPLEXTYPE DATAREQUEST 164

C.17 COMPLEXTYPE DATASPACESEXAMPLE..... 166

C.18 COMPLEXTYPE DEFAULTFILESERVERREQUEST..... 166

C.19 COMPLEXTYPE DETAILEDBEHAVIOURSTATE..... 166

C.20 COMPLEXTYPE DOWNLOADLIST 167

C.21 COMPLEXTYPE EGSCCACTIVITYREPORT 167

C.22 COMPLEXTYPE EGSCCACTIVITYREPORTLISTELEMENT 167

C.23 COMPLEXTYPE FARCCATALOGUEREQUEST 167

C.24 COMPLEXTYPE INTERRUPTPART 168

C.25 COMPLEXTYPE JOBSFILTER 169

C.26 COMPLEXTYPE JOBIDPART 169

C.27 COMPLEXTYPE JOBIDPARTLIST 170

C.28 COMPLEXTYPE JOBLISTIDPART 170

C.29 COMPLEXTYPE LASTCONSOLIDATIONREQUEST 170

C.30 COMPLEXTYPE LISTALIASINFO 170

C.31 COMPLEXTYPE LISTSTRING 171

C.32 COMPLEXTYPE PACKETSTATISTICSRESPONSE 171

C.33 COMPLEXTYPE PARAM..... 172

C.34 COMPLEXTYPE PARAMDEFINITION 173

C.35 COMPLEXTYPE PARAMDEFINITIONLIST 173

C.36 COMPLEXTYPE PARAMDEFINITIONLISTELEMENT..... 174

C.37 COMPLEXTYPE PARAMDEFINITIONRESPONSE..... 174

C.38 COMPLEXTYPE PARAMPACKETINSTANCE 175

C.39 COMPLEXTYPE PARAMPACKETINSTANCES..... 176

C.40 COMPLEXTYPE PARAMPREVIEW 176

C.41 COMPLEXTYPE PARAMPREVIEWLIST 177

C.42 COMPLEXTYPE PARAMPREVIEWLISTELEMENT..... 177

C.43 COMPLEXTYPE PARAMPREVIEWRESPONSE 177

C.44 COMPLEXTYPE PARAMREPRESENTATIONSAMPLE 178

C.45 COMPLEXTYPE PARAMREPRESENTATIONSAMPLES 178

C.46 COMPLEXTYPE PARAMRESPONSE 179

C.47 COMPLEXTYPE PARAMSAMPLELIST 179

C.48 COMPLEXTYPE PARAMSAMPLELISTELEMENT 181

C.49 COMPLEXTYPE PARAMSTATISTICLISTELEMENT..... 182

C.50 COMPLEXTYPE PARAMSTATISTICS..... 182

C.51 COMPLEXTYPE PARAMSTATISTICSLIST 183

C.52 COMPLEXTYPE PARAMSTATISTICSRESPONSE 183

C.53 COMPLEXTYPE PKTEV 184

C.54 COMPLEXTYPE PKTEVRAW..... 185

C.55 COMPLEXTYPE PKTEVSTATISTICS 186

C.56 COMPLEXTYPE PKTRAWRESPONSE..... 186

C.57 COMPLEXTYPE PKTRAWRESPONSEELEMENT 187

C.58 COMPLEXTYPE PKTC 187

C.59 COMPLEXTYPE PKTCRAW 188

C.60 COMPLEXTYPE PKTCSTATISTICS..... 189

C.61 COMPLEXTYPE PKTM..... 190

C.62 COMPLEXTYPE PKTMGAPREPORT 191

C.63 COMPLEXTYPE PKTMGAPREPORTELEMENT..... 192

C.64 COMPLEXTYPE PKTMRAW 193

C.65 COMPLEXTYPE PKTMSTATISTICS..... 194

C.66 COMPLEXTYPE REQUESTMESSAGEPART 194

C.67 COMPLEXTYPE RESPONSE..... 195

C.68 COMPLEXTYPE RESPONSEDATA..... 196

C.69 COMPLEXTYPE RESPONSEDATAELEMENT..... 196

C.70 COMPLEXTYPE RESPONSEPART..... 196

C.71 COMPLEXTYPE RESUMEPART..... 197

C.72 COMPLEXTYPE RESUMEPARTLIST 197

C.73 COMPLEXTYPE SMONPARAM 198

C.74 COMPLEXTYPE SMONPARAMRESPONSE 198

C.75 COMPLEXTYPE SMONPARAMRESPONSELIST..... 198

C.76 COMPLEXTYPE SMONPARAMRESPONSELISTELEMENT 199

C.77 COMPLEXTYPE SMONPARAMSAMPLELIST 199

C.78 COMPLEXTYPE SMONPARAMSAMPLELISTELEMENT 199

C.79 COMPLEXTYPE STATISTICSRECORD..... 200

C.80 COMPLEXTYPE SUSPENDPART..... 200

C.81	COMPLEXTYPE SUSPENDPARTLIST	201
C.82	COMPLEXTYPE TRANSFORMATIONSList	201
C.83	COMPLEXTYPE TRANSFORMATIONSREQUEST	201
C.84	COMPLEXTYPE VALIDITY.....	201
C.85	SIMPLETYPE ALIASINTERNALTYPE	202
C.86	SIMPLETYPE DELTABEHAVIOURSTATE.....	202
C.87	SIMPLETYPE DIRECTION.....	202
C.88	SIMPLETYPE FILTERTYPE.....	202
C.89	SIMPLETYPE LIMITBEHAVIOURSTATE	203
C.90	SIMPLETYPE SCCBEHAVIOURSTATE	203
C.91	SIMPLETYPE STATUSBEHAVIOURSTATE	203
C.92	SIMPLETYPE VALIDITYVALUE.....	203
C.93	COMPLEXTYPE EGSCCLOGMESSAGEREPORT.....	204
C.94	COMPLEXTYPE EGSCCRESTLOGMESSAGEREPORT.....	205
C.95	COMPLEXTYPE EGSCCRESTPKTTMREPORT	206
C.96	COMPLEXTYPE EGSCCRESTPKTTCREPORT	207
C.97	COMPLEXTYPE EGSCCRESTPARAM.....	207
C.98	COMPLEXTYPE EGSCCRESTPKTTM.....	208
C.99	COMPLEXTYPE EGSCCRESTACTIVITYREPORT	209
C.100	COMPLEXTYPE EGSCCRESTOOLREPORT	210

APPENDIX D COMMON DATA TYPES.....211

D.1	COMPLEXTYPE APIDLIST	211
D.2	COMPLEXTYPE COMPLETION.....	211
D.3	COMPLEXTYPE CONTEXTPART.....	211
D.4	COMPLEXTYPE DATASOURCEACCESS	212
D.5	COMPLEXTYPE DELIVERYRANGE.....	212
D.6	COMPLEXTYPE DOMAINLIST	212
D.7	COMPLEXTYPE EMAILLIST	213
D.8	COMPLEXTYPE PARAMNAMELIST.....	213
D.9	COMPLEXTYPE PACKETNAME	213
D.10	COMPLEXTYPE PACKETNAMELIST.....	214
D.11	COMPLEXTYPE Pi1LIST	214
D.12	COMPLEXTYPE Pi2LIST	214
D.13	COMPLEXTYPE REQUESTPOSTPROCESSING	215
D.14	COMPLEXTYPE RESPONSEFILES	215
D.15	COMPLEXTYPE SPIDLIST.....	215
D.16	COMPLEXTYPE SUBTYPELIST.....	215
D.17	COMPLEXTYPE TIME	216
D.18	COMPLEXTYPE TIMECHOICE	216
D.19	COMPLEXTYPE TIMERANGE	217
D.20	COMPLEXTYPE TIMEWINDOW	218
D.21	COMPLEXTYPE TIMEWINDOWWITHLIMIT.....	218
D.22	COMPLEXTYPE USER.....	218
D.23	SIMPLETYPE ACCOUNTREQUESTSUBTYPE.....	218
D.24	SIMPLETYPE DATAACCESSDATAELEMENT	219
D.25	SIMPLETYPE DATACOMPRESSION	220
D.26	SIMPLETYPE DATAENCRYPTING.....	220
D.27	SIMPLETYPE DATASOURCE.....	220
D.28	SIMPLETYPE DIRECTION.....	220
D.29	SIMPLETYPE DOYDATETIME.....	221
D.30	SIMPLETYPE NAMEPREFIXSUFFIXELEMENT.....	221
D.31	SIMPLETYPE BEHAVIOURSTATE	221
D.32	SIMPLETYPE OOLBEHAVIOURDELTASTATE	222
D.33	SIMPLETYPE OOLBEHAVIOURLIMITSTATE.....	222
D.34	SIMPLETYPE OOLBEHAVIOURSCCSTATE.....	222
D.35	SIMPLETYPE OOLBEHAVIOURSTATUSSTATE.....	222
D.36	SIMPLETYPE OOLPARAMETERSTATE	223
D.37	SIMPLETYPE OOLRECORDTYPE.....	223
D.38	SIMPLETYPE OOLSTATE.....	223

D.39 SIMPLETYPE OPERATION..... 224

D.40 SIMPLETYPE ORDERBY 224

D.41 SIMPLETYPE PARAMVALIDITYSTATUS 224

D.42 SIMPLETYPE PRIVACYTAG..... 225

D.43 SIMPLETYPE SPIDSSPECIFIER 225

D.44 SIMPLETYPE STATE 225

D.45 SIMPLETYPE MESSAGESEVERITY 226

D.46 SIMPLETYPE MESSAGE TYPE 226

D.47 SIMPLETYPE MESSAGEACKNOWLEDGEMENTTYPE..... 227

D.48 SIMPLETYPE ALARMSTATE..... 227

D.49 SIMPLETYPE PROCESSINGSTATE..... 227

D.50 SIMPLETYPE VALIDITYTYPE 228

D.51 SIMPLETYPE ACTIVITYSTATE 228

D.52 SIMPLETYPE PRIORITYLEVEL 229

APPENDIX E DELIVERY DATA TYPE..... 230

E.1 COMPLEXTYPE ACKDELIVERY 230

E.2 COMPLEXTYPE DELIVERY 230

E.3 COMPLEXTYPE EMAILDELIVERY..... 230

E.4 COMPLEXTYPE FILESERVERDELIVERY 231

E.5 COMPLEXTYPE SERVERDELIVERY 232

E.6 COMPLEXTYPE TARGETLOCATION..... 232

E.7 SIMPLETYPE FILENAME 233

E.8 SIMPLETYPE TIMEFORMAT 233

APPENDIX F FILE SYSTEM DATA TYPES..... 234

F.1 COMPLEXTYPE FILEFILTERENTRY 234

F.2 COMPLEXTYPE FILESYSTEMCATALOGUE 234

F.3 COMPLEXTYPE FILESYSTEMCATALOGUEFORMAT 235

F.4 COMPLEXTYPE FILESYSTEMFILE 235

F.5 COMPLEXTYPE FILESYSTEMFILECATALOGUE 236

F.6 COMPLEXTYPE FILESYSTEMFILECATALOGUEELEMENT 237

F.7 COMPLEXTYPE FILESYSTEMFILECATALOGUERESPONSE 237

F.8 COMPLEXTYPE FILESYSTEMFOLDERCATALOGUE 238

F.9 COMPLEXTYPE FILESYSTEMFOLDERCATALOGUEELEMENT 238

F.10 COMPLEXTYPE FILESYSTEMFOLDERCATALOGUERESPONSE 238

F.11 COMPLEXTYPE FILESYSTEMSUBSCRIPTION..... 239

F.12 COMPLEXTYPE FILESYSTEMSUBSCRIPTIONNOTIFICATION..... 240

F.13 COMPLEXTYPE FILESYSTEMDELIVERY 240

APPENDIX G FILTER DATA TYPE 241

G.1 COMPLEXTYPE ARCHIVEFILEFILTER 241

G.2 COMPLEXTYPE ARCHIVEFILELIST..... 241

G.3 COMPLEXTYPE ARCHIVEFILELISTELEMENT..... 242

G.4 COMPLEXTYPE CATALOGUEFILTER 242

G.5 COMPLEXTYPE CATALOGUEFILTERLIST..... 242

G.6 COMPLEXTYPE CATALOGUEFILTERLISTELEMENT..... 243

G.7 COMPLEXTYPE EDDSUSAGEREPORTFILTER..... 243

G.8 COMPLEXTYPE EDDSUSAGEREPORTFILTERELEMENT..... 244

G.9 COMPLEXTYPE EDDSUSAGEREPORTFILTERKEYWORD 244

G.10 COMPLEXTYPE EGSCCACTIVITYFILTER..... 244

G.11 COMPLEXTYPE EGSCCPARAMDEFINITIONFILTER 245

G.12 COMPLEXTYPE EGSCCPARAMTMFILTER..... 245

G.13 COMPLEXTYPE EGSCCPKTTcFILTERKEYWORD..... 245

G.14 COMPLEXTYPE EGSCCPKTTmFILTERKEYWORD..... 245

G.15 COMPLEXTYPE EGSCCTcFILTERELEMENT 245

G.16 COMPLEXTYPE EGSCCTcFILTERLIST..... 245

G.17 COMPLEXTYPE EGSCCTcPACKETFILTER..... 245

G.18 COMPLEXTYPE EGSCCTmFILTERELEMENT 246

G.19 COMPLEXTYPE EGSCCTmFILTERLIST..... 246

G.20	COMPLEXTYPE EGSCCTMPACKETFILTER	246
G.21	COMPLEXTYPE EVFILTERELEMENT	246
G.22	COMPLEXTYPE EVFILTERLIST	246
G.23	COMPLEXTYPE EVPACKETFILTER	247
G.24	COMPLEXTYPE OOLDATAREPORTFILTER	247
G.25	COMPLEXTYPE OOLFILTERELEMENT	247
G.26	COMPLEXTYPE OOLFILTERKEYWORD	249
G.27	COMPLEXTYPE OOLFILTERLIST	250
G.28	COMPLEXTYPE PARAMFILTERKEYWORD	250
G.29	COMPLEXTYPE PARAMFILTERLIST	251
G.30	COMPLEXTYPE PARAMFILTERLISTELEMENT	251
G.31	COMPLEXTYPE PARAMPREVIEWFILTER	251
G.32	COMPLEXTYPE PARAMSTATISTICSFILTER	251
G.33	COMPLEXTYPE PARAMSTREAMFILTERKEYWORD	252
G.34	COMPLEXTYPE PARAMSTREAMFILTERLIST	252
G.35	COMPLEXTYPE PARAMSTREAMFILTERLISTELEMENT	252
G.36	COMPLEXTYPE PARAMTMFILTER	253
G.37	COMPLEXTYPE PKTEVFILTERKEYWORD	254
G.38	COMPLEXTYPE PKTTCFILTERKEYWORD	255
G.39	COMPLEXTYPE PKTTMFILTERKEYWORD	256
G.40	COMPLEXTYPE PKTTMGAPREPORTFILTER	257
G.41	COMPLEXTYPE RAWEVPACKETFILTER	257
G.42	COMPLEXTYPE RAWPACKETFILTER	258
G.43	COMPLEXTYPE RAWTCPACKETFILTER	258
G.44	COMPLEXTYPE RAWTMPACKETFILTER	259
G.45	COMPLEXTYPE SMONPARAMFILTER	259
G.46	COMPLEXTYPE SMONPARAMFILTERKEYWORD	260
G.47	COMPLEXTYPE SMONPARAMFILTERLIST	260
G.48	COMPLEXTYPE SMONPARAMFILTERLISTELEMENT	260
G.49	COMPLEXTYPE TCFILTERELEMENT	260
G.50	COMPLEXTYPE TCFILTERLIST	261
G.51	COMPLEXTYPE TCPACKETFILTER	261
G.52	COMPLEXTYPE TMFILTERELEMENT	261
G.53	COMPLEXTYPE TMFILTERLIST	262
G.54	COMPLEXTYPE TMPACKETFILTER	262
G.55	SIMPLETYPE COMMANDSOURCE	263
G.56	SIMPLETYPE REPRESENTATIONSELECTION	263
G.57	SIMPLETYPE DARCREPRESENTATIONSELECTION	263
G.58	SIMPLETYPE STAGESTATUS	263
G.59	SIMPLETYPE TC TIMEFILTERING	264
G.60	SIMPLETYPE ACTIVITYTIMEFILTERING	264
G.61	SIMPLETYPE TIMEFILTERING	265
G.62	SIMPLETYPE UPLINKMODE	265
G.63	SIMPLETYPE SEVERITY	265
G.64	SIMPLETYPE CATEGORY	265
G.65	COMPLEXTYPE EGSCCLOGMESSAGEFILTER	266
G.66	COMPLEXTYPE MESSAGEFILTERLIST	266
G.67	COMPLEXTYPE MESSAGEFILTERLISTELEMENT	266
G.68	COMPLEXTYPE MESSAGEFILTERKEYWORD	267
G.69	COMPLEXTYPE EGSCCRESTLOGMESSAGEFILTER	268
G.70	COMPLEXTYPE EGSCCRESTM_PACKETFILTER	268
G.71	COMPLEXTYPE EGSCCRESTTCPACKETFILTER	269
G.72	COMPLEXTYPE EGSCCRESTTCFILTERLIST	269
G.73	COMPLEXTYPE EGSCCRESTTCFILTERELEMENT	269
G.74	COMPLEXTYPE EGSCCRESTPKTTCFILTERKEYWORD	270
G.75	COMPLEXTYPE EGSCCPARAMTMFILTER	271
G.76	COMPLEXTYPE EGSCCRESTACTIVITYFILTER	272
G.77	COMPLEXTYPE EGSCCRESTPARAMTMFILTER	272
G.78	COMPLEXTYPE EGSCCRESTM_FILTERLIST	273
G.79	COMPLEXTYPE EGSCCRESTM_FILTERELEMENT	273

G.80	COMPLEXTYPE EGSCCRESTPKTTMFILTERKEYWORD.....	273
G.81	COMPLEXTYPE EGSCCRESTPARAMFILTERLIST	274
G.82	COMPLEXTYPE EGSCCRESTPARAMFILTERELEMENT.....	274
G.83	COMPLEXTYPE EGSCCRESTPARAMFILTERKEYWORD.....	274
G.84	COMPLEXTYPE EGSCCRESTACTIVITYFILTERLIST	274
G.85	COMPLEXTYPE EGSCCRESTACTIVITYFILTERELEMENT.....	275
G.86	COMPLEXTYPE EGSCCRESTACTIVITYFILTERKEYWORD	275
G.87	SIMPLETYPE EGSCCTIMEFILTERING.....	275
G.88	COMPLEXTYPE EGSCCRESTOOLFILTERLIST	276
G.89	COMPLEXTYPE EGSCCRESTOOLFILTERELEMENT.....	276
G.90	COMPLEXTYPE EGSCCRESTOOLFILTERKEYWORD.....	276

APPENDIX H FORMAT DATA TYPE.....278

H.1	GROUP SPLITTABLEFORMAT	278
H.2	COMPLEXTYPE CATALOGUEFORMATTING.....	278
H.3	COMPLEXTYPE EDDSUSAGEREPORTFORMATTING	278
H.4	COMPLEXTYPE EVPACKETFORMATTING	279
H.5	COMPLEXTYPE PACKETFORMATTING	279
H.6	COMPLEXTYPE PACKETSTATISTICSFORMATTING	279
H.7	COMPLEXTYPE PARAMDEFINITIONFORMATTING.....	280
H.8	COMPLEXTYPE PARAMPREVIEWFORMATTING	280
H.9	COMPLEXTYPE PARAMSTATISTICSFORMATTING	280
H.10	COMPLEXTYPE PARAMTMFORMATTING.....	281
H.11	COMPLEXTYPE PARCRAWPACKETFORMATTING.....	281
H.12	COMPLEXTYPE PKTTMGAPREPORTFORMAT.....	281
H.13	COMPLEXTYPE REPORTFORMATTING	282
H.14	COMPLEXTYPE SMONPARAMFORMATTING	282
H.15	COMPLEXTYPE EGSCCACTIVITYFORMATTING	282
H.16	SIMPLETYPE ARCHIVEFILEFORMAT.....	283
H.17	SIMPLETYPE CATALOGUEFORMAT	283
H.18	SIMPLETYPE EDDSUSAGEREPORTFORMAT	283
H.19	SIMPLETYPE EVPACKETFORMAT	283
H.20	SIMPLETYPE PACKETFORMAT	283
H.21	SIMPLETYPE PACKETSTATISTICSFORMAT	284
H.22	SIMPLETYPE PARAMDEFINITIONFORMAT	284
H.23	SIMPLETYPE PARAMPREVIEWFORMAT	284
H.24	SIMPLETYPE PARAMSTATISTICSFORMAT	285
H.25	SIMPLETYPE PARAMTMFORMAT.....	285
H.26	SIMPLETYPE PARCRAWPACKETFORMAT.....	285
H.27	SIMPLETYPE REPORTFORMAT	285
H.28	SIMPLETYPE SMONPARAMFORMAT	286
H.29	SIMPLETYPE EGSCCACTIVITYFORMAT	286
H.30	COMPLEXTYPE EGSCCLOGMESSAGEFORMATTING.....	286
H.31	SIMPLETYPE EGSCCLOGMESSAGEFORMAT.....	287
H.32	COMPLEXTYPE EGSCCRESTLOGMESSAGEFORMATTING.....	287
H.33	SIMPLETYPE EGSCCRESTLOGMESSAGEFORMAT.....	287
H.34	SIMPLETYPE EGSCCRESTPARAMFORMAT	288
H.35	COMPLEXTYPE EGSCCRESTPARAMFORMATTING	288

APPENDIX I REPORT DATA TYPE.....289

I.1	COMPLEXTYPE ACTIVITYDETAILS	289
I.2	COMPLEXTYPE ACTIVITYINVOKEDETAILS	289
I.3	COMPLEXTYPE DATAREQUESTBASE	289
I.4	COMPLEXTYPE EDDSUSAGEREPORT.....	290
I.5	COMPLEXTYPE EDDSUSAGEREPORTLIST	290
I.6	COMPLEXTYPE EDDSUSAGEREPORTLISTELEMENT.....	291
I.7	COMPLEXTYPE EDDSUSAGEREPORTRESPONSE	292
I.8	COMPLEXTYPE EGSCCACTIVITYREPORT	292
I.9	COMPLEXTYPE EGSCCACTIVITYREPORTLIST.....	292
I.10	COMPLEXTYPE EGSCCACTIVITYREPORTLISTELEMENT	292

I.11	COMPLEXTYPE EGSCCACTIVITYREPORTRESPONSE.....	292
I.12	COMPLEXTYPE EGSCCPKTTcREPORT.....	292
I.13	COMPLEXTYPE EGSCCPKTTcREPORTLIST	293
I.14	COMPLEXTYPE EGSCCPKTTcREPORTLISTELEMENT	293
I.15	COMPLEXTYPE EGSCCPKTTcREPORTRESPONSE.....	293
I.16	COMPLEXTYPE EGSCCPKTTmREPORT	293
I.17	COMPLEXTYPE EGSCCPKTTmREPORTLIST.....	293
I.18	COMPLEXTYPE EGSCCPKTTmREPORTLISTELEMENT	293
I.19	COMPLEXTYPE EGSCCPKTTmREPORTRESPONSE.....	294
I.20	COMPLEXTYPE EVENTRECORDREPORT	294
I.21	COMPLEXTYPE EVENTRECORDREPORTLIST.....	294
I.22	COMPLEXTYPE EVENTRECORDREPORTLISTELEMENT	295
I.23	COMPLEXTYPE EVENTRECORDREPORTRESPONSE.....	295
I.24	COMPLEXTYPE OOLDATAREPORTLIST.....	296
I.25	COMPLEXTYPE OOLDATAREPORTRESPONSE.....	296
I.26	COMPLEXTYPE OOLDATAREPORTRESPONSEELEMENT.....	297
I.27	COMPLEXTYPE OOLRECORDREPORT	298
I.28	COMPLEXTYPE PKTTcREPORT	299
I.29	COMPLEXTYPE PKTTcREPORTLIST.....	299
I.30	COMPLEXTYPE PKTTcREPORTLISTELEMENT	300
I.31	COMPLEXTYPE CUSTOMFIELD.....	302
I.32	COMPLEXTYPE PKTTcREPORTPARAMETERLIST.....	302
I.33	COMPLEXTYPE PKTTcREPORTPARAMETERLISTELEMENT.....	303
I.34	COMPLEXTYPE PKTTcREPORTRESPONSE	303
I.35	COMPLEXTYPE PKTTmREPORT	304
I.36	COMPLEXTYPE PKTTmREPORTLIST.....	304
I.37	COMPLEXTYPE PKTTmREPORTLISTELEMENT.....	305
I.38	COMPLEXTYPE PKTTmREPORTPARAMETERLIST	306
I.39	COMPLEXTYPE PKTTmREPORTPARAMETERLISTELEMENT.....	306
I.40	COMPLEXTYPE PKTTmREPORTRESPONSE	307
I.41	COMPLEXTYPE PKTTmGAPREPORTLIST	307
I.42	COMPLEXTYPE PKTTmGAPREPORTLISTELEMENT	307
I.43	COMPLEXTYPE PKTTmGAPREPORTRESPONSE.....	308
I.44	COMPLEXTYPE EGSCCLOGMESSAGELISTELEMENT	309
I.45	COMPLEXTYPE EGSCCLOGMESSAGELIST	310
I.46	COMPLEXTYPE EGSCCLOGMESSAGERESPONSE.....	310
I.47	COMPLEXTYPE XMLMCMPARAMETERUPDATE	310
I.48	COMPLEXTYPE MCMPARAMETERUPDATES.....	310
I.49	COMPLEXTYPE EGSCCPARAMETERRESPONSE.....	310
I.50	COMPLEXTYPE EGSCCOOLPARAMETERRESPONSE	311

APPENDIX J SCHEDULE DATA TYPE312

J.1	COMPLEXTYPE ENDTIME	312
J.2	COMPLEXTYPE ONCESCHEDULE	312
J.3	COMPLEXTYPE REPEATINGCHEDULE	313
J.4	COMPLEXTYPE REPETITION	313
J.5	COMPLEXTYPE SCHEDULE	313
J.6	COMPLEXTYPE SCHEDULEONCE	314
J.7	COMPLEXTYPE SCHEDULEREPEATING	315
J.8	COMPLEXTYPE SCHEDULEWITHEXPIRY	315

APPENDIX K USER MANAGEMENT DATA TYPE.....316

K.1	COMPLEXTYPE ACCOUNTENABLEDDETAILS.....	316
K.2	COMPLEXTYPE ACCOUNTREQUEST.....	316
K.3	COMPLEXTYPE ACCOUNTREQUESTMESSAGEPART	317
K.4	COMPLEXTYPE CONTACTDETAILSELEMENT	317
K.5	COMPLEXTYPE CONTACTDETAILSLIST.....	318
K.6	COMPLEXTYPE DATAACCESSDATA.....	318
K.7	COMPLEXTYPE DATAACCESSSET	318
K.8	COMPLEXTYPE DATAACCESSLIST	318

K.9	COMPLEXTYPE GROUP	319
K.10	COMPLEXTYPE GROUPDATA	320
K.11	COMPLEXTYPE GROUPROLEASSIGNMENT.....	320
K.12	COMPLEXTYPE MISSION.....	320
K.13	COMPLEXTYPE MISSIONDATA.....	321
K.14	COMPLEXTYPE MISSIONDETAILSLIST.....	321
K.15	COMPLEXTYPE MISSIONSLIST	321
K.16	COMPLEXTYPE OPERATIONSET	322
K.17	COMPLEXTYPE OPERATIONSETDATA	322
K.18	COMPLEXTYPE QUOTASET	322
K.19	COMPLEXTYPE QUOTASETDATA	323
K.20	COMPLEXTYPE QUOTASETROLEASSIGNMENT	324
K.21	COMPLEXTYPE ROLE.....	324
K.22	COMPLEXTYPE ROLESLIST	324
K.23	COMPLEXTYPE USERACCOUNT	325
K.24	COMPLEXTYPE USERACCOUNTBASE	326
K.25	COMPLEXTYPE USERACCOUNTCREATE	328
K.26	COMPLEXTYPE USERACCOUNTDETAILS.....	330
K.27	COMPLEXTYPE USERATTRIBUTES.....	331
K.28	COMPLEXTYPE USERMISSION	331
K.29	COMPLEXTYPE USERMISSIONELEMENT	332
K.30	COMPLEXTYPE USERQUOTADETAILS	332
K.31	COMPLEXTYPE USERQUOTADETAILSLIST.....	332
K.32	COMPLEXTYPE USERROLE.....	333
K.33	COMPLEXTYPE USERSLIST.....	333
K.34	SIMPLETYPE OPERATIONELEMENT.....	333
K.35	SIMPLETYPE USEROPERATIONELEMENT	334
APPENDIX L STREAM REQUEST DATA TYPE.....		335
L.1	COMPLEXTYPE OOLSTREAM	335
L.2	COMPLEXTYPE PARAMSTREAM.....	335
L.3	COMPLEXTYPE PKTEVSTREAM	336
L.4	COMPLEXTYPE PKTSTREAM	336
L.5	COMPLEXTYPE PKTTCSTREAM	336
L.6	COMPLEXTYPE PKTTMSTREAM.....	337
L.7	COMPLEXTYPE STREAMDATAREQUEST	337
L.8	COMPLEXTYPE STREAMREQUEST	338
L.9	COMPLEXTYPE STREAMREQUESTMESSAGEPART	338
APPENDIX M EDDS PARAMETER SPREADSHEET.....		339
M.1	TDRS LIKE SPREADSHEET.....	339
M.2	PRESTO PLOT STYLESHEET	339
APPENDIX N GOOGLE PROTOCOL BUFFERS.....		340
N.1	INTRODUCTION	340
APPENDIX O CODE EXAMPLES		341
O.1	PARAMETER DEFINITION DOWNLOADER	341
O.2	DARC PARAMETER DOWNLOADER.....	347
APPENDIX P EDDS DEVOPS		357

TABLE OF TABLES

TABLE 1 - DARC STREAMING SUPPORT.....	7
TABLE 2 - PACKET TM.....	7
TABLE 3 - PACKET TC.....	8
TABLE 4 - PACKET EV	8
TABLE 5 - OUT OF LIMITS	8

TABLE 6 - COMMON REQUEST FIELDS 12

TABLE 7 - TM PACKET REQUEST..... 14

TABLE 8 - TC PACKET HISTORY 15

TABLE 9 - TC PACKET BRIEF AND FULL REPORT..... 16

TABLE 10 - ARCHIVEFILE REQUEST 16

TABLE 11 - DATA TYPE & SERVICE OVERVIEW 17

TABLE 12 - REQUEST AND RESPONSE MAPPINGS 20

TABLE 13 - DATA TYPE FORMATTING 23

TABLE 14 - STRUCTURE OF EDDS BINARY DATA 23

TABLE 15 - FORMAT OF THE EDDS FILE HEADER 24

TABLE 16 - TM PACKET RAW HEADER FIELDS..... 25

TABLE 17 - TM PACKET RAW DATA ELEMENT..... 26

TABLE 18 - POSSIBLE TC PACKET RAW HEADER FIELDS 28

TABLE 19 - TC PACKET RAW DATA ELEMENT 28

TABLE 20.1 - POSSIBLE TC PACKET RAW HEADER FIELDS 29

TABLE 21 - POSSIBLE EV PACKET RAW HEADER FIELDS..... 30

TABLE 22 - EV PACKET RAW DATA ELEMENT 30

TABLE 23 - EXAMPLE TM PACKET BINARY FORMAT 31

TABLE 24 - EXAMPLE TC PACKET BINARY FORMAT 33

TABLE 25 - EDDS USAGE REPORT 34

TABLE 26 - TC REPORT ASCII COLUMN OUTPUT..... 37

TABLE 27 - DECODING FILES..... 40

TABLE 28 - REQUEST ID CONVENTION 67

TABLE 29 - RESPONSE ID CONVENTION 67

TABLE 30 - INSTANTIATING 133

TABLE 31 - SESSION MANAGEMENT 133

TABLE 32 - ISSUING REQUESTS..... 133

TABLE 33 - MANAGING REQUESTS 134

TABLE 34 - SPECIAL REQUESTS 134

TABLE 35 - UTILITY METHODS FOR REQUESTS 135

TABLE 36 - UTILITY METHODS FOR USER MANAGEMENT..... 135

TABLE 37 - EDDS CLIENT MAVEN DEPENDENCIES..... 145

TABLE 38 - EDDS PUSH NOTIFICATIONS CLIENT..... 148

TABLE 39 - LIST OF THE PROTO FILES WITHIN EDDS 340

TABLE OF FIGURES

FIGURE 1 - INTERFACE OVERVIEW 10

FIGURE 2 - FILTER EXPRESSIONS 14

FIGURE 3 - MAVEN PROJECT SETUP 142

FIGURE 4 - MAVEN PROJECT SETUP ARCHETYPE..... 143

FIGURE 5 - MAVEN PROJECT SETUP: GROUP AND ARTIFACT ID..... 143

FIGURE 6 - ADD JAVA NATURE TO A MAVEN PROJECT..... 144

FIGURE 7 - PROJECT COMPILER COMPLIANCE LEVEL 144

1. Introduction

1.1 Purpose

This document defines the service interfaces available to clients of the EGOS Data Dissemination System (EDDS).

Note: In most cases, an XML schema file is associated to the interface. In the case of updates, the XML files shall be the first source of update. Only after the update of the XML schema file, the change can be reported back to the EUD document.

1.2 Scope

This document describes the service interfaces between the EDDS and its clients. The following interfaces are described:

- EDDS Web Services Interface
- Stream Web Service Interface

This document is a description of the interfaces detailing the common elements that are to be found for all missions. It is expected that other documentation will be available to describe mission-specific elements of the interfaces described.

The readership is expected to be software developers, ESA technical authorities with interest in the EDDS component and users of the EDDS.

It is expected that mission specific Data Definition Interface Documents (DDID) will be created based on the information given in this document (EUICD).

1.3 Document Overview

Section 1 - Introduction (this section) provides the purpose, scope and this document's overview.

Section 2 - References provides the list of reference documents.

Section 3 - Glossary provides a list of acronyms and terms used throughout this document.

Section 4 - Software Overview provides a brief description of the software system and its context.

Section 5 – Describes the data types that are supported by the interface.

Section 6 – Describes the formatting of the data provided through the interface.

Section 7 - The output is a binary file of the continuous raw data as provided by RawBodyData attribute. Naming Conventions

Section 8 – Describes the Delivery mechanism provided by the EDDS.

Section 9 - Describes the underlying web service operations that provide service functionality.

Section 10 – Describes the EDDS Email interface

Section 11 – Describes the client hosted File Server interface

Section 12 – Describes the generic client Java API provided by the EDDS

Appendix A – Provides a list of XML schemas, which define the acknowledgement data types, as described in the acknowledgement.xsd.

Appendix B – Provides a list of XML schemas, which define the authentication data types, as described in the authentication.xsd.

Appendix C – Provides a list of XML schemas, which define the batch request data types, as defined in the batchRequest.xsd.

C.97 – Provides a list of XML schemas, which define the common data types, as defined in the common.xsd.

D.45 – Provides a list of XML schemas, which define the delivery data types, as defined in the delivery.xsd.

Appendix F – Provides a list of XML schemas, which define the file system data types, as defined in the filesystem.xsd

Appendix G – Provides a list of XML schemas, which define the filter data types, as defined in the filtered.xsd.

G.69 – Provides a list of XML schemas, which define the format data types, as defined in the format.xsd.

H.32 – Provides a list of XML schemas, which define the report data types, as defined in the report.xsd.

I.47 – Provides a list of XML schemas, which define the schedule data types, as defined in the schedule.xsd.

Appendix K – Provides a list of XML schemas, which define the user management data types, as defined in the userrequest.xsd.

Appendix L – Provides a list of XML schemas, which define the stream data types, as defined in the streamrequest.xsd.

Appendix M – Provides info about the EDDS Parameter Spreadsheet.

Appendix N - Google Protocol Buffers

2. References

2.1 Applicable documents

Ref.	Document Title	Issue and Revision, Date
[AD-1]	CCSDS, XML Formatted Data Unit (XFDU) Structure and Construction Rules.	BLUE BOOK 661.0-B.1, Set 2008
[AD-2]	Generic Data Disposition System (GDDS) Generic Data Delivery Interface Document DDID (EGOS-MCS-GDDS-1003) Appendix F	Issue 1.4, 2008-04-11

2.2 Reference documents

Ref.	Document Title	Issue and Revision, Date
[RD-1]	EDDS Software User Manual [EGOS-GEN-EDDS-SUM-1001]	Version 22.0, 2025-01-10
[RD-2]	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies [RFC 2045]	November 1996
[RD-3]	Web Services Definition Language (WSDL) 1.1 [http://www.w3.org/TR/wsdl]	2001-03-15
[RD-4]	MUST Architectural Design Document [no reference]	Version 1.1, 2006-01-13
[RD-5]	XSL Transforms (XSLT) Version 1.0 [http://www.w3.org/TR/xslt]	1999-11-16
[RD-6]	Extensible Markup Language (XML) 1.0 (Fourth Edition) [http://www.w3.org/TR/xml]	2006-09-29
[RD-7]	SOAP Version 1.2 Part 1: Messaging Framework [http://www.w3.org/TR/soap12-part1/]	2003-06-24
[RD-8]	Deleted	Deleted
[RD-9]	Deleted	Deleted
[RD-10]	SCOS-2000 Glossary, Definitions and Acronyms [EGOS-MCS-S2K-GLO-0001]	Version 1.7 2007-03-20
[RD-11]	Software Requirements Specification (SRS), EGOS Data Dissemination System (EDDS) [EGOS-GEN-EDDS-SRS-1001]	Version 22.0, 2021-05-31
[RD-12]	Software Design Document (SDD), EGOS Data Dissemination System (EDDS) [EGOS-GEN-EDDS-SDD-1001]	Version 24.0, 2025-01-10
[RD-13]	Generic Data Disposition System (GDDS) Generic Data Delivery Interface Document (DDID) [EGOS-MCS-GDDS-1003]	Version 1.4, 2008-04-11
[RD-14]	TDRS External Interfaces Control Document [EGOS-MCS-S2K-ICD-0017]	Version 5.1, 2008-08-22
[RD-15]	EDDS Configuration and Installation Guide [EGOS-GEN-EDDS-CIG-1001]	Version 24.0, 2025-01-10
[RD-16]	Google Protocol Buffers [https://developers.google.com/protocol-buffers/]	2012-09-17

3. Glossary

3.1 Acronyms

Acronyms	Description
AES	Advanced Encryption Standard
APID	Application Identifier
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
CUC	CCSDS Unsegmented Code
DARC	Parameter Archive
DDID	Data Definition Interface Document
DVD	Digital Video Disc (Digital Versatile Disc)
EDDS	EGOS Data Dissemination System
ESA	European Space Agency
EGOS	ESA Ground Operations System
ERT	Earth Reception Time (Note: SCOS-2000 uses the SCOS-2000 reception time stamp and not the ground station reception time stamp.)
EV	Event (Mission control system event)
FARC	File Archive
GDDS	Generic Data Disposition System
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over Secure socket layer
JMS	Java Messaging Service
MAS	Mission Automation System
MCS	Mission Control System
MMI	Man Machine Interface
MUST	Mission Utility & Support Tool
OBEV	On-board Event
OBQ	On-board Queue
OBSM	On-board Software Maintenance
OBT	On-board Time
OOL	Out Of Limit
PARC	Packet Archive
PID	Process Identifier
PUS	Packet Utilisation Standard
SCOS	Spacecraft Control and Operations System
SMTP	Simple Mail Transfer Protocol
SMTPs	Simple Mail Transfer Protocol Secure
SOAP	<i>Originally - Simple Object Access Protocol - but since version 1.2 it has no official meaning</i>
SPID	SCOS-2000 Packet Identifier
SUM	Software Users Manual
TBD	To be decided
TBW	To be written
TC	Telecommand
TM	Telemetry
WSDL	Web Services Description Language
XFDU	XML Formatted Data Unit
XML	Extensible Mark-up Language

Acronyms	Description
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

3.2 Definition of Terms

Terms	Description
Batch Service	An EDDS service that allows a user or client application to make a request for mission data and receive a (single) response from the EDDS that contains the complete set of data requested.
Stream Service	An EDDS service that allows a client application to receive a continuous stream of mission data rather than a finite stored data set.
SCOS-2000	SCOS-2000 is a generic configurable spacecraft control and monitoring system with multi-domain and multi-mission capabilities, which operates in a scalable distributed environment. It is intended to be taken by client missions and customised to meet their specific requirements. At the time of writing this document SCOS-2000 release 5 is expected to be the baseline.
Web Services	Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions thanks to the use of XML. They can be combined in a loosely coupled way in order to achieve complex operations. Programs providing simple services can interact with each other in order to deliver sophisticated added-value services. [quoted from W3C Web services Activity Statement - http://www.w3.org/2002/ws/Activity]
SCOS-2000 Domain	The controlled or controlling entity whose packet data is stored in a dedicated SCOS-2000 multi-domain archive. For example, a spacecraft or a ground station or a controlling domain such as System Control. Each individual spacecraft will be mapped to a specific domain. Each domain will have its own instance of SCOS-2000. Taken from [RD-10].
SCOS-2000 Packet Identifier (SPID)	The identifier of the history file containing all instances of a SCOS-2000 packet. This is unique for each packet type (TM, TC and EV). In the telemetry processing, the SPID is also used as the unique identifier of the packet content for fixed packets. Taken from [RD10].
SCOS-2000 DataStream	A logical grouping of SCOS-2000 history files used to partition the historical data and to enable the selective retrieval and processing of them. Also refers to a stream of telemetry transfer frame data received by the SCOS-2000 Packetiser. Taken from [RD10].
SCET	Used in this document to indicate the value of the counter from the spacecraft's onboard clock (i.e., an integer value).
Onboard Time (OBT)	Used in this document to mean the UTC correlated onboard time corresponding to SCET. It should be noted that duplicate packets with the same SCET can have a different OBT if they are time-stamped using a different time-correlation.

4. Software Overview

The primary purpose of the EDDS is to provide controlled access to mission data for users who do not have access to the mission control system (MCS) monitoring and control facilities. The data includes science and non-science telemetry data, telecommand data, and auxiliary data such as flight dynamics data.

The EDDS offers data services and management services. A user interface is provided to these services through the EDDS Client Application see [RD-1]. This document (EUICD) describes the interfaces available to users and client applications. These external interfaces provide access to the EDDS data services:

- Authentication Services
- Batch Service
- Stream Service
- User Management Service

4.1 EDDS Services Overview

4.1.1 Authentication Services

The EDDS User Management Service allows a client application to make request for the log in and log out of the users from the EDDS. It also allows for a sufficiently privileged user to update the LDAP details held on themselves or others.

4.1.2 Batch Service

The batch service is intended to allow clients to make requests for mission data and receive a data response that contains a complete set of data that matches the request. In general, a request lists the data types and allows the user to apply a set of filters for each data type. The response data is sent to the client via the delivery method indicated in the request. A batch service request can be viewed as transient in the sense that the EDDS processes the request, builds the data set (by retrieving the relevant data) and then delivers the data set to the user. The request is then considered complete.

A batch request is submitted as an XML document. The data response can take one of these forms:

- A binary or ASCII file (can also be compressed or encrypted);
- An XML file displayed on the MMI.

The format of response content depends on the type of requests as described in Section 6

Each request always has associated acknowledgement data. The acknowledgement data can be return to the user on explicit user request. The acknowledgement data indicates the status of the request and gives details for failures. Acknowledgement data is received as an e-mail or can be downloaded as an XML document.

The Batch request response files can be compressed. The compressed files can be one of a number of file archive formats that are supported by the EDDS. By default, compression is applied to the data. Currently the supported compressed file types are:

- zip – ZIP;
- tar - USTAR format (uncompressed);
- tar.gz - USTAR format compressed with gzip.

On top to the data compression, the EDDS allows to encrypt the data according to the AES algorithm, which is enabled or disabled at mission level.

4.1.3 Stream Service

An EDDS stream service allows a client application to receive a continuous stream of mission data rather than a finite stored data set. A client application makes a request for the stream service to the EDDS and then actively requests the streamed data from the EDDS.

A stream request is submitted as an XML document. The data response can be obtained either directly from the ActiveMQ topic or via the client API via CometD. The EDDS MMI uses the client API. Each message on the topic represents one packet or parameter, each associated with the original request by the request ID header field on the message. Currently only streaming of parameters from the DARC is supported.

The payload of each message contains a framed byte array, which represents an encoded parameter using Google Protocol Buffers. The payload can be decoded using the provided helper classes found in the `esa.egos.edds.types` package. For DARC Parameters, the `DarcParamRec` class can be used. The proto files describing the data structure are stored in `edds-ws-common` and are compiled as a part of the build process.

4.1.3.1 DARC Streaming Support

Receiving streamed data from the DARC only works with compatible versions of DARC for SLES 12 (version 3.0.0 or later). EDDS expects the DARC to put parameters samples onto an ActiveMQ topic owned by the DARC with the following header data:

Header Field	Type	Description
ValueType	String	The type of the value field – one of BOOLEAN, INTEGER, STRING, DOUBLE, FLOAT, LONG
Value	Variable	Value of the parameter sample. Type can be either Boolean, Short, Integer, String, Double, Float. For Short values, the ValueType should be set to INTEGER.
Validity	Integer	One of 0 (Valid), 1 (Invalid), 2 (Unknown), 3 (Expired)
SystemElement	String	Any String value
Source	String	Any String value
State	Short	Any Short value
GenericData	String	Any String value
GenerationTime	Double	Unix time stamp of the generation time
Name	String	Name of the parameter

Table 1 - DARC Streaming Support

When a user requests live parameter samples from the DARC, EDDS will add a listener to this topic with a selector on these header fields based on the filter specified in the request. This improves performance, as only matching messages are delivered by the broker to EDDS for processing. EDDS then creates a new message with two header fields, “RequestId” (with the user’s original request ID) and “MISSION_NAME” for the mission the stream relates to. It then creates a Google Protocol Buffers encoded record (See Appendix N) with the data from the header fields of the original message and adds it to the payload of a new message. These messages are received by the web server (or a 3rd party application – as they are placed on a topic, this doesn’t interfere with the web server) and passed on to any listening clients via CometD.

4.1.3.2 PARC Streaming Support

Streaming of live packet data is supported only via the Data Provision Services that can be found in SCOS 5.4.21 and 5.5.4 or later. Once a request for live streaming data is submitted (can be for TM, TC, EV (SCOS Event Log) or OOL) and the request becomes ACTIVE, then the data is put onto the ActiveMQ message bus onto a topic for any 3rd party application to listen to directly. Alternatively, if access to the message bus is not possible, then the method “startStreamData(RequestID)” on the WSDL can be executed to start streaming the live data to clients over HTTP(S) via CometD. The client must then listen to the relevant Bayeux channel as specified in the next subsections.

4.1.3.3 Packet TM

Messages are placed onto the topic “esa.egos.edds.packet.stream.tm” with the following header values:

Header Field	Type	Description
RequestId	String	The ID of the request from the client that initiated the live streaming
MISSION_NAME	String	The name of the mission that the request relates to

Table 2 - Packet TM

The body of the message is as defined for binary Packet TM Report response files and can be found in 6.7.3.3.9

Clients must call startStreamData(RequestID) on the EDDS Web Server and then listen on the Bayeux channel NotificationChannel.PKT_TM_STREAM. When finished, clients should call the stopStreamData(RequestId) method.

A code example for streaming live telemetry data can be found under edds-ws-client/src/test/java in packet esa.egos.edds.ws.client.examples.StreamLivePktTmData

4.1.3.4 Packet TC

Messages are placed onto the topic “esa.egos.edds.packet.stream.tc” with the following header values:

Header Field	Type	Description
RequestID	String	The ID of the request from the client that initiated the live streaming
MISSION_NAME	String	The name of the mission that the request relates to

Table 3 - Packet TC

The body of the message is as defined for binary Packet TC Report response files and can be found in 6.7.3.3.7

Clients must call startStreamData(RequestID) on the EDDS Web Server and then listen on the Bayeux channel NotificationChannel.PKT_TC_STREAM. When finished, clients should call the stopStreamData(RequestId) method.

4.1.3.5 Packet EV

Messages are placed onto the topic “esa.egos.edds.packet.stream.ev” with the following header values:

Header Field	Type	Description
RequestID	String	The ID of the request from the client that initiated the live streaming
MISSION_NAME	String	The name of the mission that the request relates to

Table 4 - Packet EV

The body of the message is as defined for binary Event Record Report response files and can be found in 6.7.3.3.5

Clients must call startStreamData(RequestID) on the EDDS Web Server and then listen on the Bayeux channel NotificationChannel.PKT_EV_STREAM. When finished, clients should call the stopStreamData(RequestId) method.

4.1.3.6 Out of Limits

Messages are placed onto the topic “esa.egos.edds.packet.stream.ool” with the following header values:

Header Field	Type	Description
RequestID	String	The ID of the request from the client that initiated the live streaming
MISSION_NAME	String	The name of the mission that the request relates to

Table 5 - Out of Limits

The body of the message is as defined for binary Out of Limits Record Report response files and can be found in 6.7.3.3.12

Clients must call `startStreamData(RequestID)` on the EDDS Web Server and then listen on the Bayeux channel `NotificationChannel.PKT_OOL_STREAM`. When finished, clients should call the `stopStreamData(RequestID)` method.

4.1.4 User Management Service

An EDDS User Management Service allows a client application to make request for the management of the users, which have access to the EDDS services. Only authorised users can access the User Management services.

A user management request is submitted as an XML document.

4.2 *EDDS Interface Overview*

The following section describes the interfaces to the EDDS services.

4.2.1 EDDS Web Service Interface

Provides the set of operations that can be used by client applications to inject batch requests, user management requests and receive responses. The interface uses Web Services technology [RD-3][RD-7].

4.2.2 Stream Web Service Interface

Provides a set of operations that can be used by client applications to make stream requests and receive data in response. The interface uses Web Services technology [RD-3][RD-7].

4.2.3 Email Interface

Used by the EDDS components to deliver acknowledgement data to the user.

4.2.4 Delivery Manager Interface

The Delivery Manager is used by the EDDS to delivery batch data responses to a client hosted file server.

4.3 *Client Applications Overview*

A user can interact with the EDDS by use of several client application types:

- EDDS Client Application (via generic APIs)
- Web Services Application
- Email Server
- Delivery Manager
- Request Submitter
- EDDS stream client

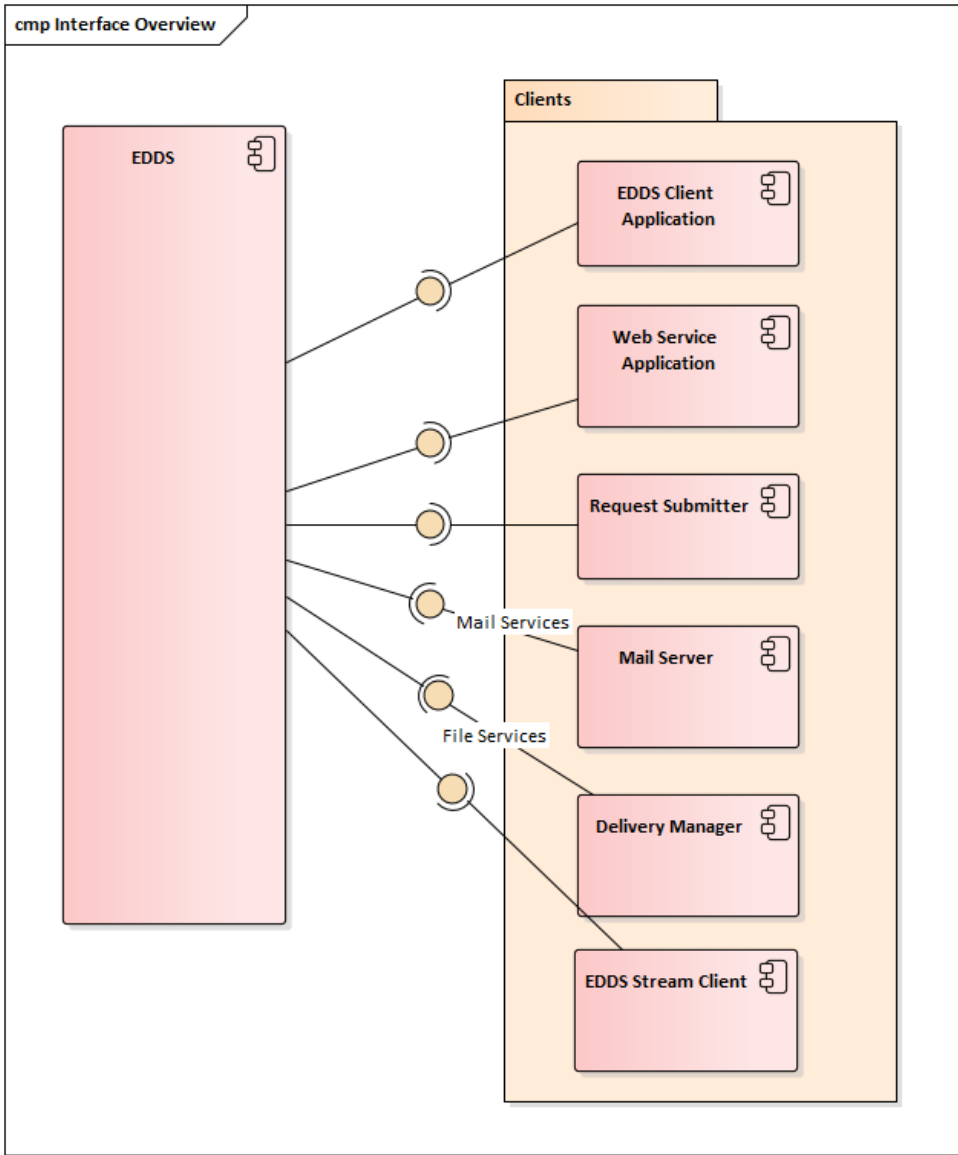


Figure 1 - Interface Overview

4.3.1 EDDS Client Application

The user interface to the EDDS is the EDDS Client Application. This application is downloaded from the EDDS web site and is run on the user’s machine. See the EDDS Client SUM [RD-1] for more information on installation configuration and usage.

The EDDS Client provides GUI access to all available EDDS services (management and batch and streaming services). This document (EUICD) describes the format of the data that is sent to the Web Services Application.

The EDDS provides access to its management batch and stream services through Web Services technology. Users can write their own client applications that access EDDS batch and stream services through the use of Web Services technology.

4.3.2 Email Server

EDDS is capable of delivering batch service acknowledgement messages to a client via email. This document (EUICD) describes the format of the acknowledgement message.

4.3.3 Delivery Manager

EDDS is capable of delivering batch service responses to a file server running on the client’s site. This document (EUICD) describes the format of batch service requests and the format of the resultant response. This document (EUICD) also describes the format of acknowledgement messages.

4.3.4 EDDS Stream client

The EDDS Stream client is a lightweight standalone application that is able to store stream data in files, in various format, from an existing edds stream request. This client is designed to be easily extensible, so that third-party users can extend the client to support new output file formats. Please refer to the CIG and SDD for further information.

4.3.5 Request Submitter

The EDDS Request Submitter is a lightweight standalone application that polls a configurable directory for request files. Any valid request files either in EDDS or GDDS format will be processed and submitted to the EDDS Server. Responses are returned as defined in the request.

All valid requests are submitted through the same user account specified in the RS configuration.

The EDDS request file must conform to the schema element RequestMessagePart as defined in [C.66]

4.3.5.1 GDDS Requests

EDDS Request Submitter also accepts requests in GDDS format [RD-13]. All GDDS request types are supported, but there are additional restrictions, because GDDS requests and EDDS requests are not fully compatible. The GDDS request is transformed into EDDS request via XSL Transformation (XSLT) and submitted for processing via EDDS client API. The mapping of GDDS schema elements is described below.

The response data will be in EDDS format and delivered by EDDS.

XSL transformation can be also performed independently of Request Submitter. On SLES 11 run from the terminal:

```
xsltproc path/to/transformation.xsl path/to/gddsRequest.xml
```

For example, to transform one of the test requests and output it to the console, you can run from edds source directory:

```
xsltproc edds-request-submitter/src/main/resources/gdds/GDDS2EDDS.xsl edds-request-submitter/src/test/resources/GDDSRequests/PktTcReportRequest.xml
```

4.3.5.1.1 Common request fields

Element in GDDS request	Is mapped to
userRequestId attribute in scheduleRequest and onlineRequest	Not used, but are added to the EDDS request comment
schedule/firstStartTime	If set, used as execution time of the first request
schedule/interval	Number of days between scheduled requests; GDDS request doesn't have end date in the schema, so 10 requests will always be created for scheduled requests
schedule/acknowledgement	Not used
general/comment	EDDS request comment field
general/userInfo/username general/userInfo/FTPpassword, general/destInfo/FTP/target	and Username and password are used as credentials for target FTP server if latter is provided; otherwise, EDDS server delivery is used.

general/destInfo/RDM and subelements	Not used
general/destInfo/Online and subelements	Not used
general/destInfo/FTP/filename	Used as the target filename
general/destInfo/FTP/directory	Target directory, if file server delivery is used
general/formatInfo/compression	EDDS compression option
general/formatInfo/SFDUrequired	Not used
general/formatInfo/missionFormat	Not used
general/dataInfo/earliestStart	If set, the execution time of the (first) request; schedule/firstStartTime overrides this value
item	Exactly one item should be specified, contains the request type and filter elements. The filter tree must not contain any OR binary operation nodes inside AND nodes

Table 6 - Common request fields

4.3.5.1.1.1 Mission and Domain

By default, the request is submitted to the default mission and domain as configured in the file poller settings. Different mission and domain can be used for the request by having extra lines in the GDDS comment field. Keyword Mission or Domain followed by a colon character, and then the EDDS mission or domain name.

Example:

```
...
    <general>
        <comment>
            Mission:TEST_MISSION
            Domain:0
            Example GDDS request comment
        </comment>
    ...
```

4.3.5.1.1.2 MultiTime elements

All the Multitime elements (i.e., SourcePktsGenTime, ExecutionTime, UplinkTime) should be in absolute time format, not in duration. Except when scheduleRequest is sent, then the required boundaries must be set as duration – start time (lower boundary) as relative time from the execution and end time as the duration after start time. Other MultiTime elements should still be in absolute time.

Example:

The following GDDS request will create scheduled request that will execute 10 requests in 5 days intervals. The packet data is retrieved starting from 3 months before the execution time and ending 21 days after start time. Additonal filter is applied with UplinkTime >= 2002-01-01T12:00:00Z.

The relevant fragments of the request:

```
<scheduleRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation='GDDSScheduleRequest.xsd'
userRequestId="20020101GDDSScheduleRequestExample">
<schedule>
    <firstStartTime>2002-01-01T20:00:00Z</firstStartTime>
    <interval>5</interval>
    <acknowledgement>true</acknowledgement>
```

```
</schedule>
<onlineRequest userRequestId="20020101GDDSDRequestExample">
...
  <item>
    <dataType>CMH</dataType>
    <dataSource>CMDH</dataSource>
    <catalogueRequest>false</catalogueRequest>
    <filter>
      <bin operation="OP_AND">
        <lhs>
          <leaf operation="OP_GTE">
            <valuePair>
              <ExecutionTime>
                <a_duration>-P3M</a_duration>
              </ExecutionTime>
            </valuePair>
          </leaf>
        </lhs>
        <rhs>
          <bin operation="OP_AND">
            <lhs>
              <leaf operation="OP_GTE">
                <valuePair>
                  <UplinkTime>
                    <a_dateTime>2002-01-01T12:00:00Z</a_dateTime>
                  </UplinkTime>
                </valuePair>
              </leaf>
            </lhs>
            <rhs>
              <leaf operation="OP_LTE">
                <valuePair>
                  <ExecutionTime>
                    <a_duration>P21D</a_duration>
                  </ExecutionTime>
                </valuePair>
              </leaf>
            </rhs>
          </bin>
        </rhs>
      </bin>
    </filter>
  </item>
</onlineRequest>
</scheduleRequest>
```

4.3.5.1.1.3 Filter expressions

Filter expressions should be simple: there should be no OR operations inside AND operations. So all the OR expressions should be towards the filter root. If there are any of those “illegal” expressions, the whole subtree will be ignored.

Example:

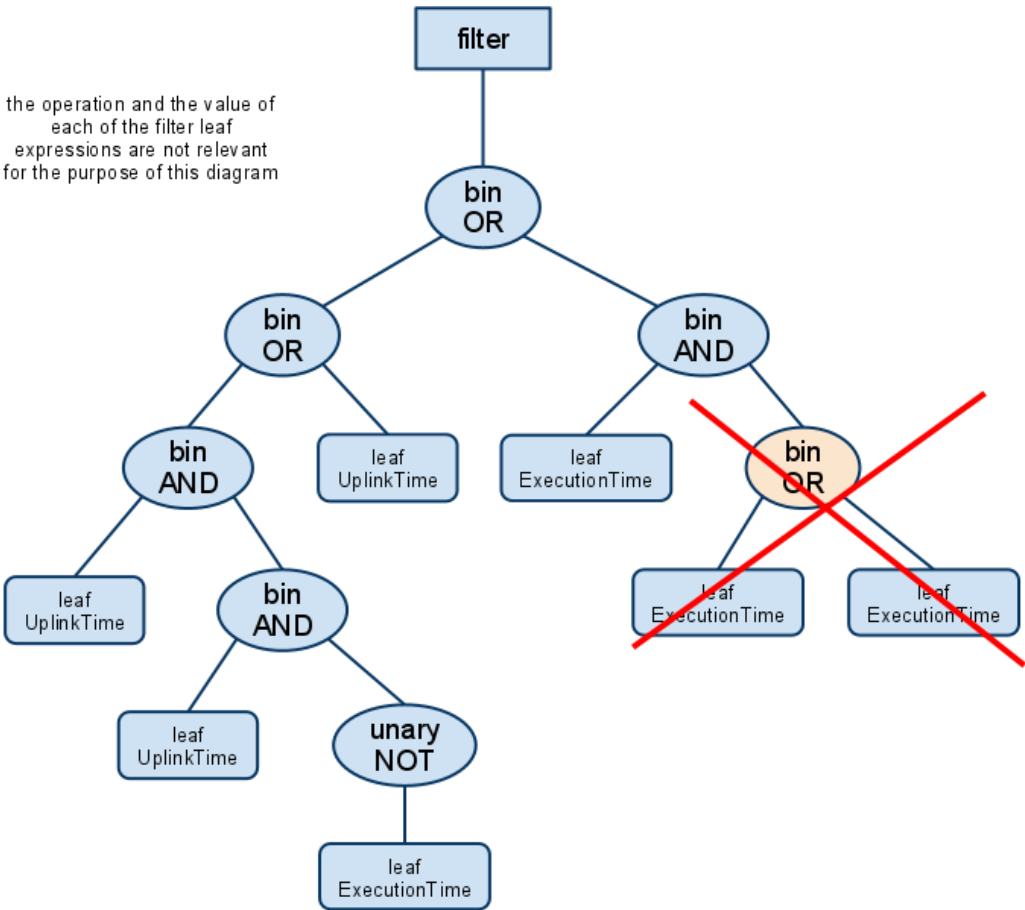


Figure 2 - Filter expressions

4.3.5.1.2 TM Packet request

This section describes the types of data request available in GDDS and how they map to the EDDS schema (where possible to map)

Item subelements	Description
dataType='TLM'	Determines the TM packet request type
dataSource	Filter packets by this Apid value
catalogueRequest	Not used
keyword	Not used
onEvent	Not used
filter	Required. Must contain a leaf element with operation OP_GTE or OP_GT and with keyword SourcePktsGenTime and another one with operation OP_LTE or OP_LT. These will be the lower and upper bound of the timerange for the query.

Table 7 - TM Packet request

Supported keywords in filter are:

Filter keyword	Mapped to	Allowed operations
SourcePktsGenTime S2KpktsGenTime	and GenerationTime	All operations
VolumeSize	MaxVolumeSize	All operations
Type, SubType, P1Val, P2Val		OP_EQ and negation in unary node with OP_NOT. One keyword can only appear once in a conjunctive clause (leafs separated with only AND expressions).

4.3.5.1.3 TC Packet History

Item subelements	Description
dataType='CMH'	Determines the TC packet request type
dataSource='CMDH'	Determines the raw TC packet request type
catalogueRequest	Not used
keyword	Not used
onEvent	Not used
filter	Required. Must contain a leaf element with operation OP_GTE or OP_GT and with keyword ExecutionTime and another one with operation OP_LTE or OP_LT. These will be the lower and upper bound of the timerange for the query.

Table 8 - TC Packet History

Supported keywords in filter are

Filter keyword	Mapped to	Allowed operations
ExecutionTime	ExecutionTime	All operations
UplinkTime	UplinkTime	All operations
VolumeSize	MaxVolumeSize	All operations

4.3.5.1.4 TC Packet brief and full report

Item subelements	Description
dataType='CMH'	Determines the TC packet request type
dataSource='CMDPB' or dataSource='CMDPF'	Whether we want to get the brief report ('CMDPB') or full report with command parameters ('CMDPF').
catalogueRequest	Not used
keyword	Not used
onEvent	Not used
filter	Required. Must contain a leaf element with keyword ExecutionTime and with operation OP_GTE or OP_GT and another one with operation OP_LTE or OP_LT. These will be the

lower and upper bound of the timerange for the query.

Table 9 - TC Packet brief and full report

Supported keywords in filter are

Filter keyword	Mapped to	Allowed operations
ExecutionTime	ExecutionTime	All operations
UplinkTime	UplinkTime	All operations
VolumeSize	MaxVolumeSize	All operations

4.3.5.1.5 ArchiveFile request

Item subelements	Description
dataType='AUX'	Determines the request type
dataSource	Not used
catalogueRequest	Not used
keyword	Not used
onEvent	Not used
filter	Required.

Table 10 - ArchiveFile request

Supported keywords in filter are

Filter keyword	Mapped to	Allowed operations
Name	Name	OP_EQ
Release, Issue	Version (Release and Issue are concatenated with a dot '.' character)	OP_EQ
CreationTime	CreationTime	Not supported
TimeSpan	ValidityTime	Not supported

5. Data Types

This section outlines a user’s perspective of the data types that the EDDS supports. Table 11 (below) gives an overview of the data types and the services which provide the data type.

Delivery Data Type		File Server	EDDS Server	Stream		Email
				Online	Offline	
Packet	TM (EDDS Raw)	✓	✓	✓	✓	-
	TM (PARC Raw)	✓	✓	✓	✓	-
	TC (EDDS Raw)	✓	✓	✓	✓	-
	TC (PARC Raw)	✓	✓	✓	✓	-
	EV (EDDS Raw)	✓	✓	✓	✓	-
	EV (PARC Raw)	✓	✓	✓	✓	-
	Statistics	✓	✓	-	-	-
Param-eter	TM	✓	✓	✓	✓	-
	Statistics	✓	✓	-	-	-
	Preview	✓	✓	-	-	-
	Definition	✓	✓	-	-	-
Report	MCS	✓	✓	-	-	-
	EDDS	✓	✓	-	-	-
Archived Files	File	✓	✓	-	-	-
	Catalogue	✓	✓	-	-	-
File System	File Catalogue	✓	✓	-	-	-
	Folder Catalogue	✓	✓	-	-	-
	File	✓	✓	-	-	-
Acknowledgement		✓	✓	-	-	✓

Table 11 - Data Type & Service Overview

5.1 Packet

Packet data provided by the EDDS consists of telemetry (TM), telecommand (TC) and MCS event (EV) packets. The packets contain binary data as processed by the MCS in Network endian order. In addition, packet statistics data is available that provides statistical information about packets stored in the MCS (e.g., the number of packets that match a given criterion for a given time span). Packets are extracted by using either the PARC Manager interface of a dedicated packet archive (PARC), the Data Provision Services , the HARC manager interface or the EGSCC REST interface. The TM/TC/EV Raw format retrieves the packet data from the PARC exactly as it is stored as a hexadecimal string, and wraps each packet in an XML tag so that they can be identified. The data is not decoded, and so cannot be filtered beyond the basics of start time, end time and (for TM and EV) SPID list.

Packet statistics data are available that provides statistical information about stored packets. Packet Reports are available that decode the encoded packet data and store the data in an XML or ASCII formatted file.

5.2 Parameter

TM parameter data can be obtained through EDDS services. Parameter data is available from a dedicated parameter archive (DARC, SMON, HARC and EGSCC).

The parameter archive is populated as TM source packets are received by the MCS and is tailored to provide fast access to the parameter data stored. Due to the large number of TM parameters defined for a mission and their associated data, it may be impractical to store all parameters in a dedicated parameter archive. The mission will be able to configure which subset of parameters is stored in the dedicated parameter archive.

The ordering of parameter samples is not guaranteed (although usually the data is ordered by sample generation time). EDDS tries to use optimal retrieval mechanisms based on the API provided by backend and does not provide any ordering on its own.

Duplicate entries might exist in special cases where request for all DARC parameter samples is being resumed (after an explicit client request to suspend/resume or after EDDS server has been stopped/killed before processing of the request was finished). To avoid duplicates, split size of the result files can be set to 0, so the entire request will be restarted from the beginning.

Parameter statistics data is available that provides statistical information about stored parameters (e.g., the number of parameters that match a given criterion for a given time span).

Parameter Definition data type is used to provide parameter static definition data (e.g., description, unit, subsystem, etc) which is derived from the spacecraft database.

5.3 Report

This data type covers both reports generated by the MCS and those generated by the EDDS. A report is always in XML, ASCII, Binary or spreadsheet format.

MCS reports can be classified into TM, TC, EV and other reports. The availability of such reports is mission dependent and includes:

- TM Packet Report
- TM Packet Gap Report
- TC Packet Report
- OOL Data Report
- Event Record Report

EDDS reports are generated by the EDDS on user request. They are:

- Status Reports (not currently supported in this version)
- Request Summary Reports (not currently supported in this version)
- System Log Reports
- EDDS Usage Report

The content of EDDS reports is given in Section 6.2.3

Reports supported by the EDDS will be in XML format, with missions providing XSLT for transforming the reports into their final delivery format. This also allows missions to specialise reports with full EDDS support.

For Out of Limit Reports, EDDS supports the following packet versions from the PARC:

- 5: For SCOS versions 5.4.19 or 5.5.2 or earlier
- 6: For SCOS versions 5.4.20 or 5.5.3 or later.
- 7: For SCOS versions 5.4.20 or 5.5.3 or later with change request GAIA-SPR-694 applied.

5.4 Archived Files

The Archived Files data type contains files stored in the MCS archive including files received from Flight Dynamics, Mission Planning and TT&C stations. Additionally, catalogue information is also available in XML format. The file data and catalogue are retrieved from the FARC. For FARC versions that support it, it is possible to register interest in items in the FARC so that when they change, a notification is received by EDDS. EDDS will then automatically retrieve the requested file from the FARC.

5.5 Acknowledgement

Acknowledgement data is data generated by EDDS and sent/taken to the user on request. Acknowledgement data contains status information. Acknowledgement data are produced by the EDDS at the different stages of the request processing and execution when the request for data is received by the EDDS or when the request has been completed successfully.

5.6 Request and Response Mappings

The table below maps the XML responses to each request so that the expected XML response types can be determined based on the request submitted. Note that not all requests will result in an XML file being returned.

Request type (BatchRequest/DataRequest)	Response type (ResponsePart/Response)
Param	ParamResponse C.46
EgscParam	EgscParameterResponse C.469
SMONParam	SMONParamResponse C.74
ParamStatistics	ParamStatisticsResponse C.52
ParamDefinition	ParamDefinitionResponse C.37
EgscParamDefinition	ParamDefinitionResponse C.37
ParamPreview	ParamPreviewResponse C.43
PktEv	N/A – binary response format
PktTm	N/A – binary response format
EgscPktTm	N/A – binary response format
PktTc	N/A – binary response format
PktEvRaw	PktRawResponse C.56
PktTmRaw	PktRawResponse C.56
EgscPktTmRaw	PktRawResponse C.56
PktTcRaw	PktRawResponse C.56
PktTmStatistics	PacketStatisticsResponse C.29
PktTcStatistics	PacketStatisticsResponse C.29
PktEvStatistics	PacketStatisticsResponse C.29
PktTmReport	PktTmReportResponse I.38
EgscPktTmReport	EgscPktTmReportResponse I.38
PktTmGapReport	PktTmGapReportResponse I.43
PktTcReport	PktTcReportResponse I.34
EgscPktTcReport	EgscPktTcReportResponse I.15
EventRecordReport	EventRecordReportResponse I.23
ArchiveCatalogue	CatalogueResponse C.14
ArchiveFile	N/A – binary response format
ArchiveSubscription	N/A – binary response format
OolRecordReport	OolDataReportResponse I.25
EddsUsageReport	EddsUsageReportResponse I.7
FileSystemFileCatalogue	FileSystemFileCatalogueResponse F.7
FileSystemFolderCatalogue	FileSystemFolderCatalogueResponse F.10
FileSystemFile	N/A – binary response format
FileSystemFileSubscription	N/A – binary response format
EgscLogMessageReport	EgscLogMessageResponse I.46
EgscRestLogMessageReport	EgscLogMessageResponse I.46
EgscRestPktTmReport	EgscPktTmReportResponse I.19
EgscRestPktTcReport	EgscPktTcReportResponse I.15
EgscRestPktTm	N/A – binary response format

Request type (BatchRequest/DataRequest)	Response type (ResponsePart/Response)
EgscRestParam	EgscParameterResponse C.46
EgscRestActivityReport	EgscActivityReportResponse I.11

Table 12 - Request and Response Mappings

6. Formats

The following section describes the formatting of data. The EDDS supports following formats:

- Binary
- EDDS Binary
- XML (and XML Transform)
- SFDU [AD-2]
- ASCII
- RawSourceBinary

Different data types support various formatting. Table 13 provides a summary of the possible formats that are supported by each request type.

Format Request Type		EDDS Binary	Binary	SFDU	XML (XML Transform)	ASCII	Others
Packet (From PARC)	TM	✓*		✓			GDDS_BINARY*
	TM (PARC Raw)		✓*		✓*		
	TM Report		✓*		✓*	✓*	RawSourceBinary*
	TM Gap Report		✓		✓		
	TM Statistics				✓		
	TC	✓*		✓			GDDS_BINARY*
	TC (PARC Raw)		✓*		✓*		
	TC Report		✓*		✓*	✓*	
	TC Statistics				✓		
	EV	✓*					
	EV (PARC Raw)		✓*		✓*		
	Event Record Report		✓*		✓*	✓*	
	EV Statistics				✓		
	OOL Record				✓*	✓*	
Packet (From HARC)	TM	✓		✓			GDDS_BINARY*
	TM (HARC Raw)		✓		✓		
	TM Report		✓		✓	✓	RawSourceBinary*
	TC Report		✓		✓	✓	RawSourceBinary*
Parameter (from SMON)	TM		✓*		✓*		TDRS
Parameter (From DARC)	TM		✓*		✓*		TDRS
	Statistics				✓*	✓*	
	Preview				✓*	✓*	
	Definition		✓*		✓*		
Parameter (From HARC)	TM		✓		✓		TDRS, DARC_BINARY*, EGSCC_BINARY*
	Definition		✓		✓		
Activity (From HARC)	Activity		✓		✓		
Log Message (From HARC)	Log Message		✓		✓		
Log Message (From EGSCC)	Log Message		✓		✓		
Parameter (From EGSCC)	TM	✓	✓		✓		TDRS, DARC_BINARY*
Packet (From EGSCC)	TM	✓		✓			GDDS_BINARY*
	TM Report		✓		✓	✓	RawSourceBinary*
	TC Report		✓		✓	✓	RawSourceBinary*

<i>Format</i> <i>Request Type</i>		<i>EDDS Binary</i>	<i>Binary</i>	<i>SFDU</i>	<i>XML (XML Transform)</i>	<i>ASCII</i>	<i>Others</i>
<i>Report (From EDDS)</i>	<i>TM Report</i>		✓		✓	✓	RawSourceBinary*
<i>Archived Files (From FARC)</i>	<i>TC Report</i>		✓		✓	✓	RawSourceBinary*
	<i>Catalogue</i>				✓	✓	
<i>File System</i>	<i>File Catalogue</i>				✓	✓	
	<i>Folder Catalogue</i>				✓	✓	
	<i>File (and Subscription)</i>		✓				
<i>Acknowledgement</i>					✓		

Table 13 - Data Type Formatting

The formats marked with * support splitting the response files into smaller chunks. The resulting files are still compliant with the format with appropriate header and tail in the files. To enable splitting, the configuration must be done by EDDS admin.

The following sections detail the formats and the applicable data types.

6.1 EDDS Binary

The format for EDDS raw data is both self-describing and extensible, and allows a mission to choose which of the standard EDDS header fields are included in that file. For a response consisting of *N* elements, the basic structure is as follows:

EDDS Raw Header
EDDS Header Fields (1)
Data Element (1)
EDDS Header Fields (2)
Data Element (2)
...
EDDS Header Fields (<i>N</i>)
Data Element (<i>N</i>)

Table 14 - Structure of EDDS Binary Data

The response data consists of a single EDDS Raw Header (based on what has been selected as the header fields), followed by the data element (this is the case both when the data is stored in a raw (ASCII hexadecimal) file, and when raw data is returned via a streaming service). Each data element is preceded by the EDDS header fields.

6.1.1 EDDS Raw Header

The EDDS Raw Header is a variable-size header, which serves the following purpose:

- Allows a user to uniquely identify binary data as being EDDS binary data
- Allows a user to find out which header fields are present in the binary data

The format of the EDDS binary header is as follows:

Field	Size (octets)	Type	Description
N	1	Integer	Total number of header fields defined for this data type. This is the total number of EDDS header fields (can be zero)
Size_1	1	Integer	Number of octets used for Header Field 1
Size_2	1	Integer	Number of octets used for Header Field 2
...	...	...	...
Size_N	1	Integer	Number of octets used for Header Field N

Table 15 - Format of the EDDS File Header

If should be underlined that in case one of the fields is a string the Size field of the header will not represent its length (since the size of a string is known only at run time) but will indicate if the field has to be included or not in the generated file.

The actual header fields, which can be present in EDDS binary data, depend on the data type; these will be defined in the following sections of this document.

Note: this section will contain the value “00” if the values of size1, size2, ..., size N (defined in the EDDS File Header) are all zero.

6.1.2 Packet Data

The packet data depends on the data type. This will be defined in the following sections for each data type. The Appendix “Packet Data” in the Configuration and Installation Guide (CIG) [RD-15] contains detailed instruction on the implementation of the header structures for the information files.

6.1.2.1 TM

The EDDS defines the following fields, which can be included in the header for TM packet data (see section 6.1.1):

Number	Field	Type	Description
1	Source Sequence Counter	Unsigned Integer	The sequential binary count of each source packet generated by an application process identified by a unique application process identifier.
2	APID	Unsigned Integer	The PUS Service Application Process Identifier, uniquely identifies the on-board source of the packet.
3	PID	Unsigned Integer	The Packet ID, calculated from the APID
4	Category	Unsigned Integer	The Packet Category, calculated from the APID
5	P1val	Unsigned Integer	SCOS-2000 P1 Value (the meaning is mission-specific)
6	P2val	Unsigned Integer	SCOS-2000 P2 Value (the meaning is mission-specific)
7	Data Stream	Unsigned Integer	SCOS-2000 Data Stream
8	Generation Time	CCSDS CUC	Time-correlated On-Board Generation Time
9	Reception Time	CCSDS CUC	Time at which the TM packet was extracted from its Transfer Frame on-ground

Number	Field	Type	Description
10	Packet Length	Unsigned Integer	Length of the packet. Added for backward compatibility with the GDDS
11	SPID	Unsigned Integer	SCOS-2000 Packet ID
12	Ground Station	Unsigned Integer	Id of the ground station which downlinked this telemetry packet
13	Virtual Channel	Unsigned Integer	Virtual Channel number of the on-board data source.
14	Data Unit type	Unsigned Integer	Identifies the data unit type
15	Type	Unsigned Integer	PUS Service Type
16	Subtype	Unsigned Integer	PUS Service Subtype
17	SleServiceID	Unsigned Integer	An 8-bit integer value that defines the SLE service used. Added for backward compatibility with the GDDS
18	Time Quality	Unsigned Integer	The value of the time quality flag 0 = GOOD, 1 = INACCURATE, 2 = BAD.
19	Quality Flag / Time Stamp Type	Unsigned Integer	Indicates the data quality of the TM packet 0 = Packet Reception (PR), 1 = Frame Transmission (FT), 2 = Packet Generation (PG)
20	Database Version	Unsigned Integer	Version of the SCOS-2000 database used to extract this packet from its transfer frame
21	Domain	Unsigned Integer	The SCOS-2000 domain in which this TM packet was generated

Table 16 - TM Packet Raw Header Fields

Note: while the data in some of these header fields may be contained in the raw TM packet data itself, these header fields are still defined by EDDS for the convenience of missions which wish to include this information also in the EDDS header for each TM packet.

For a specific telemetry packet data file, the EDDS file header determines which fields are included in that file, and the corresponding field lengths.

Note any string field will be translated in the packet as follow:

<string length in hexadecimal >+<string represented in hexadecimal >

In the case of the TM packet data type, the data field itself consists of the raw packet as generated by the spacecraft but preceded by an EDDS sequence counter and the packet length:

Field	Content	Size	Description
Count	EDDS Sequence Count	1 byte	Data element sequence count applied by EDDS. Client applications can use this to detect missing data. Note that the counter is reset to 0 once 255 is reached. If suspending and resuming a request, the counter is reset on resume. The counter continues when the file is split.
Length	Data Length	4 bytes	Size of the Data field in octets
Data	TM packet	Raw	Raw packet in network endian order

Table 17 - TM Packet Raw Data Element

6.1.2.1.1 EGSCC TM

Number	Field	Type	Description
1	APID	Unsigned Integer	The PUS Service Application Process Identifier, uniquely identifies the on-board source of the packet.
2	PID	Unsigned Integer	The Packet ID, calculated from the APID
3	Generation Time	CCSDS CUC	Time-correlated On-Board Generation Time
4	Time Quality	Unsigned Integer	The value of the time quality flag 0 = GOOD, 1 = INACCURATE, 2 = BAD, 255 = NA
5	Live	Boolean	Is live data
6	Packet Instance ID	String	Packet instance ID
7	Earth Received Time	String	Earth received time
8	Source Sequence Counter	Unsigned Integer	The sequential binary count of each source packet generated by an application process identified by a unique application process identifier.
9	PUS Type	Unsigned Integer	PUS Service Type
10	Spacecraft ID	Unsigned Integer	Spacecraft ID
11	Quality Flag / Time Stamp Type	Unsigned Integer	Indicates the data quality of the TM packet 0 = Packet Reception (PR), 1 = Frame Transmission (FT), 2 = Packet Generation (PG)
12	Virtual Channel	Unsigned Integer	Virtual Channel number of the on-board data source.
13	PUS Subtype	Unsigned Integer	PUS Service Subtype
14	Egsccl Reception Time	CCSDS CUC	Time at which the TM packet was extracted from its Transfer Frame on-ground
15	Route	String	Route
16	Data Partition	String	Data partition
17	Storage Time	String	Storage time

Table 17.1 – EGSCC TM Packet Raw Header Fields

6.1.2.2 TC

The EDDS defines the following fields, which may be included in the header for TC packet data (see section 6.1.1):

Number	Field	Type	Description
1	Type	Unsigned Integer	PUS Service Type
2	Source Type	Byte	Type of originating command source
3	Subtype	Unsigned Integer	PUS Service Subtype
4	APID	Unsigned Integer	Application Process Identifier
5	PID	Unsigned Integer	Packet ID, calculated from the APID
6	Category	Unsigned Integer	Packet Category, calculated from the APID
7	Virtual Channel	Unsigned Integer	The Virtual Channel ID
8	Verification Status	Unsigned Integer	Verification Status. This corresponds to the Command Verification Stages defined in field 13. (The meaning of the verification status is mission-specific.)
9	Uplink Time	CCSDS CUC	Time of command uplink (if any)
10	Execution Time	CCSDS CUC	Predicted/actual time of command execution
11	Last Update Time	CCSDS CUC	Time the command was last updated
12	Release Time	CCSDS CUC	Time the command was released
13	Sequence Name	String	Name of the top-level parent sequence to which this instance of the command belongs (if any)
14	Subschedule	Unsigned Integer	On-board Sub Schedule to which this command belongs
15	Subsystem	String	Associated Sub-system
16	Command Source	String	The name of the workstation used to release the telecommand
17	Command Acknowledgement	Unsigned Integer	Four-bit representation of the Service 1 report packets to be generated by the spacecraft
18	Command Verification Stages	String	The verification stages of the telecommand as supported by infrastructure (R, GTO or AC).
19	Data Stream	Unsigned Integer	SCOS-2000 Data Stream
20	Domain	Unsigned Integer	SCOS-2000 Domain
21	Database Version	String	Version of the SCOS-2000 database used to encode this telecommand

Number	Field	Type	Description
22	PUS Source Sequence Count	Unsigned Integer	PUS Source Sequence Counter
23	CUV	Unsigned Integer	Command UV Onboard verification stage
24	CAV VC0	Unsigned Integer	Command EV Acceptance verification stage
25	CEV VC0	Unsigned Integer	Command EV Execution start verification stage
26	CAV VC1	Unsigned Integer	Command EV Acceptance verification stage
27	CEV VC1	Unsigned Integer	Command EV Execution start verification stage
28	Packet Length	Unsigned Integer	Size of the packet
29	Command Name	String	The name of the command

Table 18 - Possible TC Packet Raw Header Fields

Note: while the data in some of these header fields may be contained in the raw TC packet data itself, these header fields are still defined by EDDS for the convenience of missions which wish to include this information also in the EDDS header for each TC packet.

For a specific telecommand packet data file, the EDDS file header determines which fields are included in that file, and the corresponding field widths.

Note any string field will be translated in the packet as follow:

<string length in hexadecimal >+<string represented in hexadecimal >

In the case of the TC packet data type, the data field itself consists of the raw telecommand packet sent to the spacecraft:

Field	Content	Size	Description
Count	EDDS Sequence Count	Unsigned Integer	Data element sequence count applied by EDDS. Client applications can use this to detect missing data.
Length	Data Length	Unsigned Integer	Size of the Data field in octets
Data	TC packet	Raw	Raw packet in network endian order

Table 19 - TC Packet raw Data Element

6.1.2.2.1 EGSCC TC

The EDDS defines the following fields, which may be included in the header for TC packet data (see section 6.1.1):

Number	Field	Type	Description
1	Type	Unsigned Integer	PUS Service Type
2	APID	Unsigned Integer	Application Process Identifier

Number	Field	Type	Description
3	PID	Unsigned Integer	Packet ID, calculated from the APID
4	Generation Time	String	Generation time
5	Time Quality	Unsigned Integer	The value of the time quality flag 0 = GOOD, 1 = INACCURATE, 2 = BAD, 255 = NA
6	Live	Boolean	Is Live
7	Packet Instance ID	String	Packet instance ID
8	Earth Received Time	String	Earth received time
9	Source Sequence Counter	Unsigned Integer	Source sequence counter
10	PUS Type	Unsigned Integer	PUS Service Type
11	Spacecraft ID	Unsigned Integer	Spacecraft ID
12	Quality Flag	Boolean	Quality Flag
13	Virtual Channel	Unsigned Integer	The Virtual Channel ID
14	PUS Subtype	Unsigned Integer	PUS Service Subtype
15	EGSCC Reception Time	String	EGSCC Reception time
16	Route	String	Route
17	Data Partition	String	Data partition
18	Activity Occurrence	String	Activity Occurrence
19	Activity Route	Unsigned Integer	Activity Route

Table 20.1 - Possible TC Packet Raw Header Fields

6.1.2.3 EV

EDDS defines the following fields, which can be included in the header for SCOS-2000 EV packet data (see section 6.1.1):

Number	Field	Type	Description
1	Event Type	String	Type of the event
2	Event Id	String	Event Id
3	Source	String	The source of the event
4	Category	String	Category of the event
5	Generation Time	CCSDS CUC	Generation time of the event

6	Domain	Unsigned Integer	SCOS-2000 domain
---	--------	------------------	------------------

Table 21 - Possible EV Packet Raw Header Fields

For a specific event packet data file, the EDDS file header determines which fields are included in that file, and the corresponding field widths.

Note any string field will be translated in the packet as follow:

<string length in hexadecimal >+<string represented in hexadecimal >

In the case of the EV packet data type, the data field itself consists of the raw SCOS-2000 event packet:

Field	Content	Size	Description
Count	EDDS Sequence Count	Unsigned Integer	Data element sequence count applied by EDDS. Client applications can use this to detect missing data.
Length	Data Length	Unsigned Integer	Size of the Data field in octets
Data	EV packet	Raw	Raw SCOS-2000 event packet data

Table 22 - EV Packet Raw Data Element

6.1.3 GDDS Binary File Example

6.1.3.1 TM Example

This section contains an example which shows how a GDDS TM packet binary data file would be represented using the EDDS binary file structure. The structure is based on the information contained in the GDDS ICD [RD-13]. The example provided assumes two TM packets (data elements) are present in the EDDS TM packet binary file.

GDDS telemetry packet header records contain the following fields:

- SCET (8 octets)
- Packet length (4 octets)
- Ground station ID (2 octets)
- Virtual Channel ID (2 octets)
- SLE Service (1 octet)
- Time Quality (1 octet)

These correspond to the following header EDDS header fields:

- Field 8 (Generation Time)
- Field 10 (Packet Length)
- Field 12 (Ground Station ID.)
- Field 13 (Virtual Channel)
- Field 17 (SLE Service ID)
- Field 18 (Time Quality Flag)

The following table illustrates the sequence and contents of each of the sections of the EDDS TM Packet binary file (the values are given in Hexadecimal. Note that two hecadecimal characters represent one

octet):

Packet Header Fields (packet 1)	Field_1 (Generation Time)	4E94 46BE A7B8 D000
	Field_2 (Packet Length)	0000 0050
	Field_3 (Ground Station ID.)	0000
	Field_4 (Virtual Channel)	0000
	Field_5 (SLE Service ID)	00
	Field_6 (Time Quality Flag)	00
Packet Data (packet 1)	TM packet	0000 0000 FFFF FFFF 0000 C800 0100 C8C8 0000 0012 5800 0000 2100 0000 1BB5 A0A6 42AF 0400 0000 0000 0000 0000 0000 0000 5000 0000 0000 0000 5900 0000 B0E9 9B08 E0B4 9208 FFFF FFFF 5550 4441 5445 2025
Packet Header Fields (packet 2)	Field_1 (Generation Time)	4E94 46BE 6334 3000
	Field_2 (Packet Length)	0000 0050
	Field_3 (Ground Station ID.)	0000
	Field_4 (Virtual Channel)	0000
	Field_5 (SLE Service ID)	00
	Field_6 (Time Quality Flag)	00
Packet Data (packet 2)	TM packet	0000 0000 FFFF FFFF 0000 C800 0200 C8C8 0000 0012 5800 0000 2100 0000 1B51 D154 B6AF 0400 0000 0000 0000 0000 0000 0000 5000 0000 0000 0000 5900 0000 0000 0000 4E94 5872 0006 BF8B 4E94 5872 0006 BF8B

Table 23 - Example TM Packet Binary Format

Note that the generation time is in two parts. The first part (octets 0-3) is the number of seconds since epoch (1st Jan 1970) and the second part (octets 4-7) is the number of microseconds.

6.1.3.2 TC Example

This section contains an example which shows how a GDDS TC packet binary data file would be represented using the EDDS binary file structure. The structure is based on the information contained in the GDDS ICD [RD-13]. The example provided assumes two TC packets (data elements) are present in the EDDS TC packet binary file.

GDDS telecommand packet header records contain the following fields:

- Uplink Time (8 octets)
- Execution Time (8 octets)

- Command Uplink Stage CUV (2 octets)
- Command Acceptance Stage Real-Time CAV (VC0) (2 octets)
- Command Execution Stage Real-Time CEV (VC0) (2 octets)
- Command Acceptance Stage Playback CAV (VC1) (2 octets)
- Command Execution Stage Playback CEV (VC1) (2 octets)
- Packet Length of the DDS telecommand packet body (4 octets)
- Command ID (8 octets)
- Parent Sequence ID (8 octets)

These correspond to the following header EDDS header fields:

- Field 9 (Uplink Time)
- Field 10 (Execution Time)
- Field 23 (CUV)
- Field 24 (CAV VC0)
- Field 25 (CEV VC0)
- Field 26 (CAV VC1)
- Field 27 (CEV VC1)
- Field 28 (Packet Length)
- Field 29 (Command Name)
- Field 13 (Sequence Name)

The following table illustrates the sequence and contents of each of the sections of the EDDS TC Packet binary file (the values are given in Hexadecimal. Note that two hexadecimal characters represent one octet):

Packet Header Fields (packet 1)	Field 6 (Uplink Time)	4FB6 5AD2 000E 9A94
	Field 7 (Execution Time)	4FB6 5BE4 0006 9398
	Field 22 (CUV)	0004
	Field 23 (CAV VC0)	1000
	Field 24 (CEV VC0)	0000
	Field 25 (CAV VC1)	1000
	Field 26 (CEV VC1)	0000
	Field 27 (Packet Length)	0000 000D
	Field 28 (Command Name)	5A41 4330 3330 3036
	Field 10 (Sequence Name)	0000 0000 0000 0000
Packet Data (packet 1)	TC packet	18BC C1F1 0006 1903 0600 0084 C0
Packet Header Fields (packet 2)	Field 6 (Uplink Time)	4FB6 5AD2 000E 9A95
	Field 7 (Execution Time)	4FB6 5AD2 000E 9A95
	Field 22 (CUV)	0004
	Field 23 (CAV VC0)	0020

	Field 24 (CEV VC0)	0000
	Field 25 (CAV VC1)	0020
	Field 26 (CEV VC1)	0000
	Field 27 (Packet Length)	0000 0021
	Field 28 (Command Name)	5A4D 5431 3130 3034
	Field 10 (Sequence Name)	0000 0000 0000 0000
Packet Data (packet 2)	TC packet	18FC C012 001A 190B 0400 0007 4FB6 5BE4 6E56 18BC C1F1 0006 1903 0600 0084 C0B5 39

Table 24 - Example TC Packet Binary Format

Note that the generation time is in two parts. The first part (octets 0-3) is the number of seconds since epoch (1st Jan 1970) and the second part (octets 4-7) is the number of microseconds.

6.2 XML

This section identifies the EDDS data types that support XML formatting and provides references to associated schemas (found inAppendix C)

6.2.1 Packet

- Statistics - The packet statistics data type supports XML formatting, see the PacketStatistics schema Appendix C
- TC / TM / EV PARC Raw – This data type supports XML formatting only. See the PktRawResponse schema Appendix C
- TC / TM / EV / OOL Reports – These data types supports XML formatting only. See the PktTcReportResponse, PktTmReportResponse, EventRecordReportResponse and OolDataReportResponse schemas in Appendix C respectively.

6.2.2 Parameter

- Param – The parameter data type supports XML formatting, see the Param schema Appendix C.
- Statistics - The parameter statistics data type supports XML formatting, see the ParameterStatistics schema Appendix C.
- Definition - The parameter definition data type supports XML formatting, see the ParameterDefinitionType schema Appendix C.
- Preview - The parameter preview data type supports XML formatting, see the ParameterPreview schema Appendix C
- SMON Parameter - The parameter data type supports XML formatting, see the SMONParamResponse schema Appendix C

6.2.2.1 A special note on times shown in the response files

There are a number of times shown in the output of the DARC Param response file:

- Generation time: for DARC sources, this corresponds to the parameter generation time. For SMON sources, this corresponds to the sample time (i.e., the packet generation time (filing time) plus an offset for supercommutated packets

- Storage time: for DARC sources, this corresponds to the DARC storage time. For SMON sources, this corresponds to the create time (i.e., the PARC storage time)
- Parent generation time: for DARC sources, this corresponds to the packet generation time. For SMON sources, this corresponds to the packet generation time (filing time)

6.2.3 Report

The EDDS provides access to EDDS generated reports.

6.2.3.1.1 EDDS Report

EDDS reports are generated by the EDDS on user request. All EDDS reports are in XML format and are packaged within an EDDS XML EddsReport type. The XML schema of the report type can be found in H.32. The reports that may be packaged within the EddsReport are as follows:

6.2.3.1.1.1 Request Summary Report

The XML schema AcknowledgementPart defines the structure and content of an EDDS XML request summary report type.

EDDS provides two views of Request Summary Reports: a brief and a full summary report. The full summary report displays the complete acknowledgement message, while the brief is a subset of info from the acknowledgement message.

The detailed structure of the AcknowledgementPart is defined in the Section A.1

6.2.3.1.1.2 EDDS Usage Report

The XML schema EddsUsageReport defines the structure and content of an EDDS XML usage report type. See I.4.

A usage report contains information about the usage of the EDDS services over a given period:

Field	Type	Description
Total Deleted Requests	Integer	Total number of deleted requests over the period
Total Processed Requests	Integer	Total number of processed requests over the period.
Total Failed Requests	Integer	Total number of failed requests over the period
The following is within the EddsUsageReportList element, within zero or more EddsUsageReportListElement elements.		
RequestId	String	The ID of the request
User	String	The username of the user who submitted the request
Role	String	The name of the role used for the request
Status	String	The final status of the request
SizeOfResponseFile	BigInteger	The size of the response file in bytes
SubmissionDate	Date	The date the request was submitted to the server
ExecutionDate	Date	The date the request was executed
CompletionData	Date	The date the request was completed

Table 25 - EDDS Usage Report

6.2.4 Archived Files

The EDDS provides access to catalogues and files of MCS file archives. The catalogue data type supports XML formatting. The corresponding XML schema, Catalogue, defines the structure and content of an EDDS XML catalogue data type. See C.1, C.5 and C.6.

Note: Files from an MCS archive are passed to the end user as retrieved from the MCS file archive. This implies that a file could of course be an XML document. The definition of the content of MCS files is mission-specific and is outside the scope of this ICD.

6.2.5 File System

The EDDS provides access to a designated directory on the file system of EDDS Server. The File and Folder Catalogue requests support XML formatting option.

6.2.6 Acknowledgement

An EDDS acknowledgement is an XML document. The structure and content are described in the XML schema acknowledgement.xsd. See A.1 for detailed information.

6.3 XML Transform

All the requests that support XML format also support XML transformations. Custom stylesheets can be added to the EDDS Server for transforming the XML responses. Available list of transformations can be retrieved through the EddsService.getTransformations() method in the Client API.

6.4 XFDU

The XFDU format is a schema that allows existing data to be packaged up with further information such as the original system used, the filename and a checksum. The XFDU schema is an external schema and can be found in the document “CCSDS, XML Formatted Data Unit (XFDU) Structure and Construction Rules” [AD-1]. EDDS supports a post processing option that, when selected, will create an XML file compliant with this schema.

6.5 TDRS Spreadsheet

The EDDS supports the formatting of data types (Parameters) into the TDRS spreadsheet format. The format is a text file containing tab delimited fields (columns) with new lines separating rows. See Appendix M for more information.

6.6 ASCII

The EDDS supports ASCII representation of data types. The contents of the ASCII files are similar to the equivalent XML files, but without the XML markup.

For the TC Report, the ASCII format is based on the Command History application found in SCOS 5.3/5.4. Full information on this format can be found in the SCOS 2000 Command History Operator User Manual, with the most useful information copied here for convenience.

The following snippet shows an example of this format:

Name	Description	S	D	C	G	B	IL	ST	Source	FC	TC	Q	R	GTO	A	SS	1122	CC
S2KTC007	TC(3,3)	E	E	E				SR	MS de-vm140	00	01	S	SSS	S				S
S2KCP013	Housekeeping SID	Eng						Dec		HK_SID_1								

To save space, the columns “Sequence, Domain, Release Time and Execution Time” have been removed from the output. The first line provides the column headings, and the following line is the values for a TC packet. The final indented line shows the parameters for the TC packet. These can be removed from the output by selecting “Get brief summary only” in the TC Report request.

The following table describes the column headings after “Description”:

Column	Description
--------	-------------

S	Static PTV Check State (E => Enabled, D => Disabled or O => Overridden)	
D	Dynamic PTV Check State (E => Enabled, D => Disabled or O => Overridden)	
C	CEV Check State (E => Enabled or D => Disabled)	
G	Group flag value (G => in a group, E => last in a group)	
B	Block flag value (B => in a block, E => last in a block)	
IL	Interlock Status, two characters, the first reports the interlock type and scope, the second, the interlock stage type.	
	Interlock Type and Scope	
	L	Wait-Fail Local
	G	Wait-Fail Global
	S	Wait-Fail Subsystem Local
	F	Fail-Only Local
	T	Fail-Only Subsystem Local
	Interlock Stage Type	
	R	1st UV stage: Ground Stn. Reception
	U	2nd UV stage: Uplink
	O	3rd UV stage: On Board Reception
	A	Execution Ver.: Acceptance
	C	Execution Ver.: Completion
ST	Source Type (MS => Man.Stack, AS => Auto Stack)	
Source	Source workstation ID	
FC	Frame Counter: indicating current frame count. Only applicable for CLTU AD mode	
TC	Transmission Counter: indicating the current number of (re)transmission. Only applicable for CLTU AD mode.	
Q	This column will show an asterisk (*) should any of the verification stages be anything other than “Success (S)” or not applicable (blank).	
Verification Status Headings		
	The status of each individual verification stage may have one of the following values:	
	F	Fail
	U	Unverified
	X	Unknown
	P	Pending
	T	Time-out
	S	Success
	I	Idle
	A	Assumed Passed
	C	Affected
	V	Uncertain Successful
	N	Uncertain Failed
	E	Superseded

	Blank N/A The following behaviour of a time-tagged commands verification status are extended: - If a time-tagged command is deleted prior to execution, a string “DELETED” is showed in verification stage field - If a time-tagged command is disabled at execution, EV stages are marked as “D”
R	Release from SCOS-2000 to the SLE User (NCTRS/NIS). Note that a failure for this stage may be caused by one of the following reasons: - PTV NOT OK at release time (the PTV state is visible in the CQD - PTV/CEV pane, see Section 3.1.3.3 below) - PTV OK (e.g., dynamic PTV was overridden) but no TCP/IP connection between the releaser and the NCTRS was available at release time - The command was successfully released but eventually rejected by the NCTRS e.g., because the required connection to the station equipment was not available at release time. In this case an administrative message is also received from the NCTRS and logged in the HFA
G	Ground station reception (UV 1st stage)
T	Uplink a.k.a. radiation (UV 2nd stage) and in case of Throw Event Command it corresponds to the Execution stage.
O	On-board reception a.k.a. transfer (3rd UV stage, only applicable in AD mode)
A	On-board application acceptance
SS	Execution starts – two instances of the stage are shown, one corresponding to updates from Real Time and the other corresponding to updates from Playback TM
1122	Execution step „n” – two instances of each stage are shown, one corresponding to updates from Real Time and the other corresponding to updates from Playback TM
CC	Execution completion – two instances of the stage are shown, one corresponding to updates from Real Time and the other corresponding to updates from Playback TM

Table 26 - TC Report ASCII column output

For the TM Report, the ASCII format of the TM parameters show ‘Parameter Name’, ‘Parameter Description’, ‘Unit’, “Bit Offset”, “Byte Offset”, “Filing Time”, “Length”, “Parameter Radix”, “Parameter Value”.

The Parameters will be shown in a row after two space in the beginning followed by the above Parameter Information which would be seperated by Tab spacing.

The third row of the below snippet shows an example of the TM parameter information in ASCII format:

```
Mnemonic      Description APID      PID Category      Database Version      Generation Time Datastream Domain Ground Station {
S2KTM980      MISC_PACKET: Conf. Variables packet by MISCdynamic server  980 61 4 1.0 2023-09-06T10:41:10.521396 1 (
MISC01        CMD_GBL_CEV_LDTT      Dec 0 36 2023-09-06T10:41:10.521396 80 Dec ENABLED
```

6.7 Binary

Binary format is specific to each of the request types.

6.7.1 Archive File

The file is stored as received from the mission File Archive, packaged in a TAR file.

6.7.2 File System

The files are stored as received from the file system, packaged in a TAR file.

6.7.3 Google Proto Buffers

Google Protocol Buffers binary format is used to deliver data in an efficient format. Compared to XML format it is more compact and faster to process in case of large data volumes. The proto files describing the data structure are stored in edds-ws-common and are compiled as a part of the build process. Data is encoded as a framed byte array. Helper classes are provided in project edds-ws-client in package esa.egos.edds.ws.client.binary to simplify working with binary format.

6.7.3.1 Header line

Each EDDS binary file has ASCII encoded header line included to provide simple interaction by external applications. Format of header line is simple key value pairs separated by semicolons:

```
# Key1: Value1; ... ; KeyN: ValueN \n
```

- Header line starts with # (byte value 23 in HEX) and space (byte value 20 in HEX)
- Each key value pair ends with semicolon (byte value 3B in HEX) and optional space (byte value 20 in HEX)
- Key is separated from value with colon (byte value 3A in HEX) and optional space (byte value 20 in HEX)
- Header line ends with single line feed character (byte value A0 in HEX)
- Currently supported keys are:
 - Type – data type in binary format (current supported values are Parameter and SMON Parameter)
 - Version – version of binary format

6.7.3.2 Data

Data format is based on Google Protocol Buffers (GPB, see [RD-16]). The file contains the list of serialized GPB DarcParameterBinary objects, each of them prefixed with its size in bytes, the value is in GPB varint format.

Note that Google Protocol Buffer encoded files are not indended to be decoded manually. Instead, the original Proto file describing the format in plain text should be compiled and the Google Protocol Buffers library should be used instead. The Proto file describing this format is provided in the EDDS source code within the edds-ws-common package. For more information on Google Protocol Buffers, see [RD-16].

6.7.3.3 Specific formats

6.7.3.3.1 Parameter TM

Protobuf format is used to deliver Parameter TM data efficiently. Binary is one of the available formats Parameter TM request submitters can choose from.

This format, associated with retrieval from the S2K DARC, is also available for retrieval from the EGS-CC HARC. In that case the selection of DARC_Binary is applicable.

6.7.3.3.1.1 Data

Following is the GPB DarcParameterBinary message definition:

```
import "ParamValidity.proto";

message DarcParameterBinary {

    enum Type {
```

```
        // Special type indicating that the proto type definition should
be updated to include a new type
        UNKNOWN = 0;
        BIT = 1;
        UTINYINT = 2;
        STINYINT = 3;
        USMALLINT = 4;
        SSMALLINT = 5;
        UMEDIUMINT = 6;
        SMEDIUMINT = 7;
        SINT = 8;
        UINT = 9;
        FLOAT = 10;
        DOUBLE = 11;
        STRING = 12;
        DATETIME = 13;
        JOB = 14;
        LOG = 15;
    }

    required string param_name = 1;
    required int64 gen_time = 2;
    required int64 storage_time = 3;
    required string value = 4;
    required ParamValidity validity = 5;
    required Type type = 6;

    // Following are not present in DARC v2.2.2 and earlier
optional int64      parentGenTime = 7; // SCOS TM packet generation
time
optional int64      parentId = 8; // SPID in PARC
optional Type      rawValType = 9;
optional bool      boolRawVal = 10;
optional string     strRawVal = 11;
optional double     dblRawVal = 12;
optional float      fltRawVal = 13;
optional sint64     longRawVal = 14;
}
```

The above message imports the ParamValidity definition which is specified as follows:

```
package esa.egos.edds.model;

option java_outer_classname = "ParamValidityProto";
enum ParamValidity {
    VALID = 0;
    INVALID = 1;
    UNKNOWN = 2;
    EXPIRED = 3;
}
```

6.7.3.3.1.2 **Decoding Files**

Note that it is not necessary to manually decode these files! EDDS provides an API in the edds-ws-client to decode them. It is also possible to decode the files using Google Protocol Buffers libraries for many other languages including Python, Perl and C++.

First line	# Type: Parameter; Version: 1;	Header line for the binary file format
Rest of file	Binary data	Framed Google Protocol Buffers encoded data. The encoded data is prefixed with the length of

		the message so that multiple messages can be included in one file. This functionality is provided by both the Apache ActiveMQ and the Google Protobuf implementations. Example code on how to read the data with either implementation are provided in the esa.egos.edds.ws.client.examples test package of edds-ws-client.
--	--	---

Table 27 - Decoding Files

6.7.3.3.2 Parameter TM Definitions

Protobuf format is used to deliver Parameter TM definitions data efficiently. Binary is one of the available formats Parameter TM Definitions request submitters can choose from.

6.7.3.3.2.1 Data

Following is the GPB ParamDefinitions message definition:

```
message ParamDefinitionRecord
{
    optional string name = 1;
    optional string description = 2;
    optional string unit = 3;
    optional string type = 4;
    optional bool active = 5;
}

message ParamDefinitions
{
    repeated ParamDefinitionRecord entry = 1;
}
```

6.7.3.3.3 SMON Parameter

The Protobuf format is used to deliver Parameter TM data from SMON efficiently. Binary is one of the available formats that the SMON Parameter request can choose from. It is also possible to choose the DARC Binary format, so that the output from the SMON Parameter request is compatible with programs that can read DARC Binary data files. See Section 6.7.3.3.1 for details of that format.

6.7.3.3.3.1 Data

Following is the GPB SMONParameter message definition:

```
message SMONParameterBinary
{
    required string name = 1;
    required string calib_id = 2;
    repeated PacketInstanceBinary packet_instances = 3;
    repeated RepresentationSampleBinary representation_samples = 4;

    enum ValidityValueBinary {
        UNKNOWN = 0;
        VALID = 1;
        INVALID = 3;
        EXPIRED = 4;
    }

    message PacketInstanceBinary {
        required string reception_time = 1;
        required string generation_time = 2;
        required int32 sample_offset = 3;
        required int32 sequence_counter = 4;
    }
}
```



```
    required int32 spacecraft_id = 5;
    required SMONParameterBinary.ValidityValueBinary validity = 6;
    required int32 data_stream = 7;
    required int32 domain = 8;
    required int32 packet_size = 9;
    required string ground_station = 10;
    required string packet_mnemonic = 11;
}

message RepresentationSampleBinary {
    required string representation_name = 1;
    required string sample_time = 2;
    required string value = 3;

    message Validity {

        required SMONParameterBinary.ValidityValueBinary validity = 1;
        required string validity_details = 2;

    }
    required Validity validity = 4;

    message BehaviourState {
        required string status_type = 1;

        enum GeneralBehaviourStateBinary {
            INCONSISTENT_STATUS = 0;
            NOMINAL = 1;
            OUT_OF_LIMIT = 2;
            UNCHECKED = 3;
            UNDEFINED = 4;
            VIOLATED = 5;
            WARNING = 6;
        }
        required GeneralBehaviourStateBinary ool_state = 2;

        required int32 nr_of_violations = 3;
        required string value = 4;

        message DetailedBehaviourStateBinary {
            optional LimitBehaviourStateBinary limitState = 1;
            optional StatusBehaviourStateBinary statusState = 2;
            optional SCCBehaviourStateBinary sccState = 3;
            optional DeltaBehaviourStateBinary deltaState = 4;

            enum LimitBehaviourStateBinary {
                WITHIN_RANGE = 0;
                LOW = 1;
                HIGH = 2;
                LOW_LOW = 3;
                HIGH_HIGH = 4;
            }

            enum StatusBehaviourStateBinary {
                ALLOWED = 0;
                UNEXPECTED = 1;
            }

            enum SCCBehaviourStateBinary {
                SCC_ENABLE = 0;
                SCC_UNINIT = 1;
            }
        }
    }
}
```

```

        SCC_DISABLE = 2;
        SCC_OFF = 3;
        SCC_FAIL = 4;
    }

    enum DeltaBehaviourStateBinary {
        DELTA_ALLOWED = 0;
        DELTA_EQUAL = 1;
        MAX_DELTA_LOW = 2;
        MAX_DELTA_HIGH = 3;
        MIN_DELTA_LOW = 4;
        MIN_DELTA_HIGH = 5;
    }
}
required DetailedBehaviourStateBinary detailed_state = 5;
}
repeated BehaviourState behaviour_states = 5;
}
}
```

6.7.3.3.4 EGSCC Parameter Report

The Protobuf format is used to deliver parameter report data from EGSCC (REST and HARC) efficiently. **Binary** is one of the available formats that the EGSCC Parameter Report request can choose from.

Time fields are expressed as seconds since epoch with pico-second precision, resulting in a string containing a fixed-point number with 12 decimal places.

6.7.3.3.4.1 Data

Following is the GPB `EgscParameterBinary` message definition:

```

package esa.egos.edds.model;
option java_outer_classname = "EgscParameterBinaryProto";

enum ActionRequiredPb {
    NO_ACTION_REQUIRED_DEFINED = 0;
    OUTSTANDING = 1;
    ACTION_TAKEN_OVER = 2;
    PREVIOUS_OUTSTANDING = 3;
    NO_ACTION_ = 4;
}

enum AlarmStatePb {
    NOT_CHECKED = 0;
    NOMINAL = 1;
    WARNING = 2;
    ALARM = 3;
    NOMINAL_NO_CHECK_DEFINED = 4;
    NOMINAL_CHECKS_DISABLED = 5;
    UNKNOWN_STATE = 6;
    VIOLATED = 7;
}

enum CategoryTypePb {
    ALL_ACTIVITIES = 0;
    SCHEDULED_ACTIVITIES = 1;
    GROUND_SCHEDULED_ACTIVITIES = 2;
    EXTERNALLY_SCHEDULED_ACTIVITIES = 3;
    RUNNING_ACTIVITIES = 4;
    FORCED_PARAMETERS = 5;
    INVALID_PARAMETERS = 6;
}
```

```
    ALARM_PARAMETERS = 7;
    ALL_EVENTS = 8;
}

enum CompoundTypePb {
    SIMPLE_VALUE = 0;
    OBJECT = 1;
    MATRIX = 2;
    AGGREGATE = 3;
    SEQUENCE = 4;
    ANY = 5;
    ENUM_VALUE = 6;
    REFERENCE = 7;
    DEDUCED = 8;
}

enum DataTypePb {
    INT8 = 1;
    INT16 = 2;
    INT32 = 3;
    INT64 = 4;
    UINT8 = 5;
    UINT16 = 6;
    UINT32 = 7;
    UINT64 = 8;
    FLOAT = 9;
    DOUBLE = 10;
    CHAR = 11;
    WIDE_CHAR = 12;
    BOOL = 13;
    BIT_STRING = 14;
    ASCII_STRING = 15;
    UTF16_STRING = 16;
    ABSOLUTE_TIME = 17;
    DURATION = 18;
    INTERVAL = 19;
    INVALID = 20;
    BYTE_STRING = 21;
}

enum GenerationTimeQualityPb {
    ACCURATE = 0;
    INACCURATE = 1;
    ESTIMATED = 2;
    BAD = 3;
    GROUND = 4;
}

enum GlobalStatePb {
    ALARM = 0;
    ALARM_ACTION = 1;
    DISABLED = 2;
    ERROR = 3;
    FORCED = 4;
    INCONSISTENT_STATE = 5;
    INVALID = 6;
    NO_VALUE = 7;
    NOMINAL = 8;
    NOMINAL_CHECKS_DISABLED = 9;
    NOMINAL_DEFAULT_VALUE = 10;
    NOMINAL_NO_CHECK_DEFINED = 11;
```

```
SUSPENDED = 12;
UNKNOWN_STATE = 13;
WARNING = 14;
WARNING_ACTION_ = 15;
}

enum McmObjectProcessingStatePb {
    PROCESSED = 0;
    FORCED = 1;
    DISABLED = 2;
    SUSPENDED = 3;
    NOT_RECEIVED = 4;
    INVALID_DEFINITION = 5;
    ERROR = 6;
}

enum MonitoringUpdateTypePb {
    CREATION = 0;
    DELETION = 1;
    UPDATE = 2;
    CHANGE = 3;
}

enum SccStatePb {
    OK = 0;
    FAILED = 1;
    NOT_INITIALISED = 2;
    OFF = 3;
    VIOLATED = 4;
}

enum TrustabilityPb {
    TRUSTABLE = 0;
    NOT_TRUSTABLE = 1;
}

enum ValidityTypePb {
    VALID = 0;
    UNKNOWN = 1;
    INVALID = 2;
    UNKNOWN_VALUE = 3;
}

message CategoryMarkerPb {
    optional bool last_update = 1;
    optional CategoryTypePb type = 2;
}

message DataPartitionIdPb {
    optional string dataspace_id = 1;
    optional string id = 2;
}

message McmObjectStatePb {
    optional string alias = 1;
    optional string annotation = 2;
    repeated CategoryMarkerPb categories = 3;
    required DataPartitionIdPb data_partition_id = 4;
    optional string description = 5;
    optional string egscce_reception_time = 6;
    optional string generation_time = 7;
```

```
    optional GenerationTimeQualityPb generation_time_quality = 8;
    optional MonitoringUpdateTypePb monitoring_update_type = 9;
    optional string processing_time = 10;
}

message StatusConsistencyCheckPb {
    optional string description = 1;
    optional SccStatePb scc_state = 2;
}

message AggregateDefinitionPb {
    required int64 field_count = 1;
}

message DeducedValueDefinitionPb {
}

message EnumValueDefinitionPb {
}

message ReferenceValueDefinitionPb {
}

message SequenceDefinitionPb {
    required ValueDefinitionPb element_type = 1;
}

message SimpleValueDefinitionPb {
    required DataTypePb data_type = 1;
    optional string data_unit = 2;
}

message ValueDefinitionPb {
    required CompoundTypePb compound_type = 1;
    required string name = 2;
    optional SimpleValueDefinitionPb simple_value_definition = 3;
    //ObjectValueDefinition object_value_definition = 4;
    //MatrixDefinition matrix_value_definition = 5;
    optional AggregateDefinitionPb aggregate_definition = 6;
    optional SequenceDefinitionPb sequence_value_definition = 7;
    //AnyValue any_value_definition = 8;
    optional EnumValueDefinitionPb enum_value_definition = 9;
    optional ReferenceValueDefinitionPb reference_value_definition = 10;
    optional DeducedValueDefinitionPb deduced_value_definition = 11;
}

message SimpleValuePb {
    optional double double_value = 1;
    optional float float_value = 2;
    optional int32 int32_value = 3;
    optional int64 int64_value = 4;
    optional uint32 uint32_value = 5;
    optional uint64 uint64_value = 6;
    optional sint32 sint32_value = 7;
    optional sint64 sint64_value = 8;
    optional fixed32 fixed32_value = 9;
    optional fixed64 fixed64_value = 10;
    optional sfixed32 sfixed32_value = 11;
    optional sfixed64 sfixed64_value = 12;
    optional bool bool_value = 13;
    optional string string_value = 14;
```

```
    optional bytes bytes_value = 15;
    required bool is_engineering_value = 16;
}

message AggregatePb {
    repeated SimpleValuePb simple_value = 1;
}

// nb, only support SimpleValue
message SequencePb {
    repeated SimpleValuePb simple_value = 1;
}

message EnumValuePb {
    required int64 number = 1;
    required string text = 2;
}

message McmObjectIdPb {
    required string definition_id = 1; // UUID
    required int64 occurrence_id = 2;
}

message ReferenceValuePb {
    required McmObjectIdPb mcm_object_id = 1;
}

// nb, only support SimpleValue, EnumValue
message DeducedValuePb {
    required ReferenceValuePb reference_value = 1;
    optional SimpleValuePb simple_value = 2;
    optional EnumValuePb enum_value = 3;
}

message ValuePb {
    required ValueDefinitionPb value_definition = 1;
    optional SimpleValuePb simple_value = 3;
    //ObjectValue object_value = 4;
    //Matrix matrix_value = 5;
    optional AggregatePb aggregate = 6;
    optional SequencePb sequence_value = 7;
    //AnyValue any_value = 8;
    optional EnumValuePb enum_value = 9;
    optional ReferenceValuePb reference_value = 10;
    optional DeducedValuePb deduced_value = 11;
}

message SourceDataPb {
    optional string raw_data_identifier = 1;
    required string raw_value = 2;
    required AlarmStatePb source_alarm_state = 3;
    optional string source_qualifier = 4;
    required ValidityTypePb source_validity = 5;
    required ValuePb source_value = 6;
}

message AlarmPb {
    optional string check_identifier = 1;
    required string description = 2;
    required AlarmStatePb alarm_state = 3;
}
```

```

message ParameterValuePb {
    required string update_time = 1;
    required ValuePb value = 2;
}

message StateTransitionDetailsPb {
    optional ParameterValuePb last_nominal = 1;
    optional AlarmPb previous_summary_alarm_state = 2;
    optional ParameterValuePb previous_value = 3;
    optional bool state_changed = 4;
    optional ParameterValuePb state_transition_value = 5;
}

message CheckResultPb {
    optional AlarmStatePb alarm_state = 1;
    optional bool enabled = 2;
    optional bool forced = 3;
    optional string id = 4;
    required int64 violated_samples_counter = 5;
}

message ParameterStatePb {
    optional ActionRequiredPb action_required = 1;
    optional ParameterValuePb assumed_value = 2;
    repeated CheckResultPb check_results = 3;
    optional string expected_reporting_cycle = 4;
    required GlobalStatePb global_state = 5;
    required bool is_full_state = 6;
    required bool is_outdated = 7;
    required McmObjectProcessingStatePb mcm_object_processing_state = 8;
    optional string reception_route = 9;
    optional StatusConsistencyCheckPb scc = 10;
    required SourceDataPb source = 11;
    optional StateTransitionDetailsPb state_transition_details = 12;
    optional AlarmPb summary_alarm_state = 13;
    required TrustabilityPb trustability = 14;
    required ValidityTypePb validity = 15;
    required ParameterValuePb value = 16;
    optional bool value_changed = 17;
}

message McmParameterUpdatePb {
    required ParameterStatePb update = 1;
    required McmObjectIdPb mcm_object_id = 2;
    required string mcm_path = 3;
}

message EgscParameterBinaryPb {
    required McmParameterUpdatePb parameter_update = 1;
}

```

6.7.3.3.5 Event Record Report

The Protobuf format is used to deliver packet TM data from PARC efficiently. Binary is one of the available formats that the Event Record Report request can choose from.

6.7.3.3.5.1 Data

Following is the GPB `EventRecordReport` message definition:

```
message EventRecordReportBinary
{
    optional string id = 1;
    optional string domain = 2;
    optional int64 generation_time = 3;
    optional string application = 4;
    optional string severity = 5;
    optional string type = 6;
    optional string workstation = 7;
    optional string message = 8;
}
```

6.7.3.3.6 Packet Raw

The Protobuf format is used to deliver raw packet data from PARC efficiently. Binary is one of the available formats that the Packet TC Raw, Packet TM Raw and Packet EV Raw requests can choose from.

6.7.3.3.6.1 Data

Following is the GPB PacketRaw message definition:

```
message PacketRawBinary
{
    optional string packet = 1;
    optional int64 packet_id = 2;
}
```

6.7.3.3.7 Packet TC Report

The Protobuf format is used to deliver packet TC report data from PARC efficiently. Binary is one of the available formats that the Packet TC Report request can choose from.

6.7.3.3.7.1 Data

Following is the GPB PacketTCReport message definition:

```
message PacketTCBinary
{
    optional string command_name = 1;
    optional string domain = 2;
    optional string source = 3;
    optional string description = 4;
    optional string sequence_name = 5;
    optional string release_time = 6;
    optional string uplink_time = 7;
    optional string execution_time = 8;
    optional string sub_system_id = 9;
    optional int64 sub_schedule_id = 10;
    optional int32 p_uSService_type = 11;
    optional int32 p_uSSub_service_type = 12;
    optional int32 apid = 13;
    optional int32 pid = 14;
    optional int32 vcid = 15;
    optional int32 category = 16;
    optional int32 ground_station_id = 17;
    optional int32 ssc = 18;
    optional string db_version = 19;
    optional string uplink_mode = 20;
    optional string static_ptv_check_state = 21;
    optional string dynamic_ptv_check_state = 22;
    optional string cev_check_state = 23;
    optional bool is_in_group = 24;
    optional bool is_group_end = 25;
}
```



```
optional bool is_in_block = 26;
optional bool is_block_end = 27;
optional string interlock_type = 28;
optional string interlock_stage_type = 29;
optional string src_type = 30;
optional string frame_count = 31;
optional string release_state = 32;
optional string ground_state = 33;
optional string uplink_state = 34;
optional string on_board_state = 35;
optional string on_board_acc_state = 36;
optional string exec_start_rt_state = 37;
optional string exec_start_pb_state = 38;
optional string exec_step1_rt_state = 39;
optional string exec_step1_pb_state = 40;
optional string exec_step2_rt_state = 41;
optional string exec_step2_pb_state = 42;
optional string exec_comp_state = 43;
optional string exec_comp_pb_state = 44;
optional string tt_obq_status = 45;
optional string tt_del_status = 49;
optional string tt_load_status = 50;
repeated PktTcReportParameterBinary pkt_tc_report_parameters = 46;
optional int32 trans_count = 47;
repeated MissionSpecificField mission_specific_fields = 48;
optional bytes body = 51;
optional string on_board_acc_pb_state = 52;

message PktTcReportParameterBinary {
    optional string parameter_name = 1;
    optional string parameter_description = 2;
    optional string parameter_unit = 3;
    optional string param_type = 4;
    optional string param_rep = 5;
    optional bool fix_edit = 6;
    optional bool has_changed = 7;
    optional string radix = 8;
    optional bool is_editable = 9;
    optional bool manually_edited = 10;
    optional string param_value = 11;
    optional string parameter_long_description = 12;
}

message MissionSpecificField {
    optional string name = 1;
    optional string type = 2;
    optional string value = 3;
}
}
```

6.7.3.3.8 EGSCC Packet TC Report

The Protobuf format is used to deliver packet TC report data from EGSCC (REST and HARC) efficiently. Binary is one of the available formats that the EGSCC Packet TC Report request can choose from.

Time fields are expressed as seconds since epoch with pico-second precision, resulting in a string containing a fixed-point number with 12 decimal places.

6.7.3.3.8.1 Data

Following is the GPB EgscPacketTCReport message definition:

```
message EgscPacketTCBinary
```

```
{
  optional int64 apid = 1;
  optional string packet_id = 2;
  optional string generation_time = 3;
  optional TimeQuality time_quality = 4;
  optional bool live = 5;
  optional string packet_instance_id = 6;
  optional string earth_received_time = 7;
  optional int64 source_sequence_counter = 8;
  optional int64 pus_type = 9;
  optional int64 spacecraft_id = 10;
  optional bool quality_flag = 11;
  optional int64 virtual_channel_id = 12;
  optional int64 pus_sub_type = 13;
  optional string egscv_reception_time = 14;
  optional int64 route = 15;
  optional bytes body = 16;
  optional string datapartition = 17;
  optional string activity_occurrence = 18;
  optional string activityroute = 19;
  repeated AnnotationPb annotations = 20;
  optional int64 transaction_id = 21;
  optional string storagetime = 22;
  optional string transmission_mode = 23;

  enum TimeQuality {
    ACCURATE = 0;
    INACCURATE = 1;
    ESTIMATED = 2;
    BAD = 3;
    NA = 255;
  }

  message AnnotationPb{
    optional string annotation_text = 1;
  }
}
```

6.7.3.3.9 Packet TM Report

The Protobuf format is used to deliver packet TM report data from PARC efficiently. Binary is one of the available formats that the Packet TM Report request can choose from.

6.7.3.3.9.1 Data

Following is the GPB PacketTMReport message definition:

```
message PacketTMBinary
{
  optional int32 apid = 1;
  optional int32 pid = 2;
  optional int32 category = 3;
  optional string database_version = 4;
  optional string generation_time = 5;
  optional string data_stream = 6;
  optional string partition_name = 25;
  optional string domain = 7;
  optional string ground_station = 8;
  optional string quality_flag = 9;
  optional string reception_time = 10;
  optional string sle_service_id = 11;
  optional int32 source_sequence_counter = 12;
```

```

optional int64 spid = 13;
optional int32 type = 14;
optional int32 sub_type = 15;
optional string time_quality = 16;
optional string virtual_channel = 17;
optional string data_unit_type = 18;
optional int32 pi1 = 19;
optional int32 pi2 = 20;
optional int32 packet_length = 21;
optional string mnemonic = 22;
optional string description = 23;
optional sint64 id = 24;
optional int32 spacecraftId = 26;
optional bytes body = 28;

repeated PktTmReportParameterBinary pkt_tm_report_parameters = 27;

message PktTmReportParameterBinary {
    optional string parameter_name = 1;
    optional string parameter_description = 2;
    optional string parameter_units = 3;
    optional string parameter_radix = 4;
    optional int64 filing_time = 5;
    optional int32 bit_offset = 6;
    optional int32 byte_offset = 7;
    optional string parameter_value = 8;
    optional int64 length = 9;
}

enum TimeQuality {
    GOOD = 0;
    INACCURATE = 1;
    BAD = 2;
    NA = 255;
}
}

```

6.7.3.3.10 EGSCC Packet TM Report

The Protobuf format is used to deliver packet EGSCC TM report data from EGSCC (REST and HARC) efficiently. Binary is one of the available formats that the EGSCC Packet TM Report request can choose from.

6.7.3.3.10.1 *Time fields are expressed as seconds since epoch with pico-second precision, resulting in a string containing a fixed-point number with 12 decimal places.Data*

Following is the GPB `EgscPacketTmReport` message definition:

```

message EgscPacketTmBinary
{
    optional int64 apid = 1;
    optional string packet_id = 2;
    optional string generation_time = 3;
    optional TimeQuality time_quality = 4;
    optional bool live = 5;
    optional string packet_instance_id = 6;
    optional string earth_received_time = 7;
    optional int64 source_sequence_counter = 8;
    optional int64 pus_type = 9;
    optional int64 spacecraft_id = 10;
    optional bool quality_flag = 11;
    optional int64 virtual_channel_id = 12;
}

```

```
optional int64 pus_sub_type = 13;
optional string egscce_reception_time = 14;
optional int64 route = 15;
optional bytes body = 16;
optional string datapartition = 17;
optional string storagetime = 18;
optional int64 transaction_id = 19;
optional string reception_route = 20;
repeated AnnotationPb annotations = 21;
optional string provider_id = 22;
optional string antenna_id = 23;
optional bool idle_frame = 24;
optional int64 master_channel_frame_counter = 25;
optional int64 master_channel_id = 26;
optional int64 virtual_channel_frame_counter = 27;
optional int64 data_link_continuity = 28;

enum TimeQuality {
  ACCURATE = 0;
  INACCURATE = 1;
  ESTIMATED = 2;
  BAD = 3;
  NA = 255;
}

message AnnotationPb{
  optional string annotation_text = 1;
}
}
```

6.7.3.3.11 Packet TM Gap Report

The Protobuf format is used to deliver packet TM gap report data from PARC efficiently. Binary is one of the available formats that the Packet TM Gap Report request can choose from.

6.7.3.3.11.1 Data

Following is the GPB PacketTMGapBinary message definition:

```
message PacketTMGapBinary
{
  optional int64 start_gap_time = 1;
  optional int64 end_gap_time = 2;
  optional int32 data_partition = 3;
  optional int32 apid = 4;
  optional int32 start_ssc_pkt = 5;
  optional int32 end_ssc_pkt = 6;
  optional int32 num_pkts_missing = 7;
}
```

6.7.3.3.12 Out of Limit Record Report

The Protobuf format is used to deliver packet OOL report data from PARC efficiently. Binary is one of the available formats that the OOL Report request can choose from.

6.7.3.3.12.1 Data

Following is the GPB OOLReport message definition:

```
package esa.egos.edds.model;

enum OOLRecordType {
  NO_VALUE = 0;
  LIMIT = 1;
  STATE = 2;
}
```

```
    STATUS_CONSISTENCY = 3;
    SOFT_HARD_LIMIT = 4;
    LIMIT_MO = 5;
    DELTA = 6;
}

message OOLReportBinary {
    message OOLReportRecord {
        enum OOLParameterState {
            NOMINAL = 0;
            WARNING = 1;
            OUT_OF_LIMITS = 2;
            SCC = 3;
            SCC_DISABLE = 4;
            SCC_OFF = 5;
            SCC_UNINIT = 6;
            VIOLATION = 7;
            NO_VALUE = 8;
        }

        enum OOLBehaviourLimitState {
            LIMIT_HIGH = 0;
            LIMIT_HIGH_HIGH = 1;
            LIMIT_LOW = 2;
            LIMIT_LOW_LOW = 3;
            LIMIT_WITHIN_RANGE = 4;
        }

        enum OOLBehaviourSccState {
            B_SCC_ENABLE = 0;
            B_SCC_DISABLE = 1;
            B_SCC_FAIL = 2;
            B_SCC_OFF = 3;
            B_SCC_UNINIT = 4;
        }

        enum OOLBehaviourDeltaState {
            DELTA_ALLOWED = 0;
            DELTA_EQUAL = 1;
            MAX_DELTA_HIGH = 2;
            MAX_DELTA_LOW = 3;
            MIN_DELTA_HIGH = 4;
            MIN_DELTA_LOW = 5;
        }

        enum OOLBehaviourStatusState {
            STATUS_ALLOWED = 0;
            STATUS_UNEXPECTED = 1;
        }

        optional OOLRecordType record_type = 1;
        optional int64 generation_time = 2;
        optional int64 parameter_changed_time = 3;
        optional string parameter_name = 4;
        optional OOLParameterState parameter_state = 5;
        optional string ool_state = 6;
        optional string value = 7;
        optional string lower_limit = 8;
        optional string upper_limit = 9;
        optional string allowed_values = 10;
        optional string scc_limit = 11;
```

```

        optional string reference_value = 12;
        optional string delta_value = 13;
        optional string min_value = 14;
        optional string last_value = 15;
        optional int32 violation_number = 16;
        optional int32 violation_number_hard = 17;
        optional int32 violation_number_soft = 18;
        optional string description = 19;
        optional string unit = 20;
        optional OOLBehaviourLimitState behaviour_limit_state = 21;
        optional OOLBehaviourScsState behaviour_scc_state = 22;
        optional OOLBehaviourDeltaState behaviour_delta_state = 23;
        optional OOLBehaviourStatusState behaviour_status_state = 24;
        optional string violation_description = 25;
    }

    repeated OOLReportRecord records = 1;
}

```

6.7.3.3.13 EGSCC Activity Report

The Protobuf format is used to deliver packet EGSCC Activity report data from EGSCC (HARC and REST) efficiently. Binary is one of the available formats that the EGSCC Activity Report request can choose from.

Time fields are expressed as seconds since epoch with pico-second precision, resulting in a string containing a fixed-point number with 12 decimal places.

6.7.3.3.13.1 Data

Following is the GPB `EgscActivity` message definition:

```

message EgscActivityBinary{

    optional ActivityInvokeDetails activity_invoke_details = 1;
    optional ActivityDetails activity_details = 2;
}

message ActivityInvokeDetails {

    optional string alias = 1;
    optional ActivityAnnotation annotation = 2;
    repeated Category category = 3;
    optional DataPartitionId data_partition_id = 4;
    optional string description = 5;
    optional string egsc_reception_time = 6;
    optional string generation_time = 7;
    optional GenerationTimeQuality generation_time_quality = 8;
    optional MonitoringUpdateType monitoring_update_type = 9;
    optional string processing_time = 10;
    optional MetaData meta_data = 11;
    repeated ActivityArgumentData activity_argument_data = 12;
    optional ActivityState assumed_state = 13;
    optional ActivityState confirmed_state = 14;
    optional string earliest_execution_start_time = 15;
    optional string external_schedule_information = 16;
    optional string latest_execution_end_time = 17;
    optional PriorityLevel priority_level = 18;
    optional string request_id = 19;
    optional string selected_route = 20;
    optional string processor_id = 21;
    optional string source_id = 22;
    repeated VerificationStage verification_stage = 23;
}

```

```
message ActivityDetails {
    optional string definition_id = 1;
    optional int64 occurrence_id = 2;
    optional string path = 3;
    optional DefinitionVersion definition_version = 4;
    optional int64 transaction_id = 5;
}

message Category {
    optional bool last_update = 1;
    optional CategoryType type = 2;
}

message DefinitionVersion{
    optional string version_tag = 0;
    optional OnlineModificationStatus online_modification_status = 1;
}

message OnlineModificationStatus{
    optional string session_name = 0;
    optional int32 update_number = 1;
    optional string update_time = 2;
}

message VerificationStage {
    optional string name = 1;
    optional ActivityState associated_state = 2;
    optional string error_reason = 3;
    optional bool is_final_failure = 4;
    optional VerificationStageState real_time = 5;
    optional VerificationStageState play_back = 6;
}

message ActivityAnnotation {
    optional string text = 0;
}

message MetaData {
    optional string key = 0;
    repeated string value = 1;
}

message DataPartitionId{
    optional DataspaceId data_space_id = 0;
}

message DataspaceId{
    optional string id = 0;
}

enum MonitoringUpdateType {
    CREATION = 0;
    DELETION = 1;
    UPDATE = 2;
    CHANGE = 3;
}

enum ActivityState {
    NOT_INITIATED = 0;
    PRECONDITIONS = 1;
```

```
PRECONDITIONS_SCHEDULED = 2;
ROUTING = 3;
ROUTING_SCHEDULED = 4;
ROUTING_WAIT_EVENT = 5;
EXECUTING = 6;
EXECUTING_PAUSING = 7;
EXECUTING_PAUSED = 8;
EXECUTING_RESUMING = 9;
COMPLETED_SUCCESS = 10;
COMPLETED_FAILURE = 11;
COMPLETED_TIMEOUT = 12;
COMPLETED_ABORTED = 13;
COMPLETED_STOPPED = 14;
CONFIRMATION_PENDING = 15;
CONFIRMATION_SUCCESS = 16;
CONFIRMATION_TIMEOUT = 17;
CONFIRMATION_FAILURE = 18;
COMPLETED_UNVERIFIED = 19;
CANCELLED_DELETED = 20;
CANCELLED_DISABLED = 21;
}

enum AssociatedState {
    ROUTING = 0;
    ROUTING_SCHEDULED = 1;
    ROUTING_WAIT_EVENT = 2;
    EXECUTING = 3;
}

enum VerificationStageState {
    NOT_APPLICABLE = 0;
    IDLE = 1;
    PENDING = 2;
    OPEN = 3;
    FAILED = 4;
    TIMEOUT = 5;
    SUCCEEDED = 6;
    UNVERIFIED = 7;
    AFFECTED = 8;
    UNCERTAIN_SUCCEEDED = 9;
    UNCERTAIN_FAILED = 10;
}

enum CategoryType {
    ALL_ACTIVITIES = 0;
    SCHEDULED_ACTIVITIES = 1;
    GROUND_SCHEDULED_ACTIVITIES = 2;
    EXTERNALLY_SCHEDULED_ACTIVITIES = 3;
    RUNNING_ACTIVITIES = 4;
    FORCED_PARAMETERS = 5;
    INVALID_PARAMETERS = 6;
    ALARM_PARAMETERS = 7;
    ALL_EVENTS = 8;
}

enum PriorityLevel {
    EXPEDITED = 0;
    STANDARD = 1;
}

enum GenerationTimeQuality {
```



```
    ACCURATE = 0;
    INACCURATE = 1;
    ESTIMATED = 2;
    BAD = 3;
    GROUND = 4;
}

message ActivityArgumentData {
    optional string name = 1;
    optional bool is_default = 2;
    optional bool has_default = 3;
    optional bool is_fixed = 4;
    optional ValuePb rawValue = 5;
    optional ValuePb engValue = 6;
    optional ValuePb sourceValue = 7;
}

message ValuePb {
    optional ValueDefinitionPb value_definition = 1;
    optional SimpleValuePb simple_value = 2;
    optional AggregatePb aggregate = 3;
    optional SequencePb sequence_value = 4;
    optional EnumValuePb enum_value = 5;
    optional ReferenceValuePb reference_value = 6;
    optional DeducedValuePb deduced_value = 7;
}

message ValueDefinitionPb {
    optional CompoundTypePb compound_type = 1;
    optional string name = 2;
    optional SimpleValueDefinitionPb simple_value_definition = 3;
    optional AggregateDefinitionPb aggregate_definition = 6;
    optional SequenceDefinitionPb sequence_value_definition = 7;
    optional EnumValueDefinitionPb enum_value_definition = 9;
    optional ReferenceValueDefinitionPb reference_value_definition =
10;
    optional DeducedValueDefinitionPb deduced_value_definition = 11;
}

enum CompoundTypePb {
    SIMPLE_VALUE = 0;
    OBJECT = 1;
    MATRIX = 2;
    AGGREGATE = 3;
    SEQUENCE = 4;
    ANY = 5;
    ENUM_VALUE = 6;
    REFERENCE = 7;
    DEDUCED = 8;
    SELECTION = 9;
}

message SimpleValueDefinitionPb {
    optional DataTypePb data_type = 1;
    optional string data_unit = 2;
}

enum DataTypePb {
    INT8 = 1;
    INT16 = 2;
    INT32 = 3;
```

```
    INT64 = 4;
    UINT8 = 5;
    UINT16 = 6;
    UINT32 = 7;
    UINT64 = 8;
    FLOAT = 9;
    DOUBLE = 10;
    CHAR = 11;
    WIDE_CHAR = 12;
    BOOL = 13;
    BIT_STRING = 14;
    ASCII_STRING = 15;
    UTF16_STRING = 16;
    ABSOLUTE_TIME = 17;
    DURATION = 18;
    INTERVAL = 19;
    INVALID = 20;
    BYTE_STRING = 21;
}

message AggregateDefinitionPb {
    optional int64 field_count = 1;
}

message DeducedValueDefinitionPb {
}

message EnumValueDefinitionPb {
}

message ReferenceValueDefinitionPb {
}

message SequenceDefinitionPb {
    optional ValueDefinitionPb element_type = 1;
}

message SimpleValuePb {
    optional double double_value = 1;
    optional float float_value = 2;
    optional int32 int32_value = 3;
    optional int64 int64_value = 4;
    optional uint32 uint32_value = 5;
    optional uint64 uint64_value = 6;
    optional sint32 sint32_value = 7;
    optional sint64 sint64_value = 8;
    optional fixed32 fixed32_value = 9;
    optional fixed64 fixed64_value = 10;
    optional sfixed32 sfixed32_value = 11;
    optional sfixed64 sfixed64_value = 12;
    optional bool bool_value = 13;
    optional string string_value = 14;
    optional bytes bytes_value = 15;
    optional bool is_engineering_value = 16;
}

message AggregatePb {
    repeated SimpleValuePb simple_value = 1;
```

```

    }

    message SequencePb {
        repeated SimpleValuePb simple_value = 1;
    }

    message EnumValuePb {
        optional int64 number = 1;
        optional string text = 2;
    }

    message ReferenceValuePb {
        optional McmObjectIdPb mcm_object_id = 1;
    }

    message DeducedValuePb {
        optional ReferenceValuePb reference_value = 1;
        optional SimpleValuePb simple_value = 2;
        optional EnumValuePb enum_value = 3;
    }

    message McmObjectIdPb {
        optional string definition_id = 1;
        optional int64 occurrence_id = 2;
    }

```

6.7.3.3.14 EGSCC Log Message Report

The Protobuf format is used to deliver packet EGSCC Log Message report data from EGSCC (HARC and REST) efficiently. Binary is one of the available formats that the EGSCC Log Message Report request can choose from.

Time fields are expressed as seconds since epoch with pico-second precision, resulting in a string containing a fixed-point number with 12 decimal places.

6.7.3.3.14.1 Data

Following is the GPB `EgscclLogMessage` message definition:

```

package esa.egos.edds.model;

option java_outer_classname = "EgscclLogMessageProto";

message EgscclLogMessageBinary {

    optional string id = 1;
    optional string logbook_id = 2;
    optional string node_id = 3;
    optional string generation_time = 4;
    optional string egsccl_reception_time = 5;
    optional string storage_time = 6;
    optional string category_id = 7;
    optional string application_id = 8;
    optional SeverityPb severity = 9;
    optional string system_id = 10;
    optional string text = 11;
    optional MessageTypePb type = 12;
    optional string trace_scope = 13;
    optional int64 trace_level = 14;

```

```
optional string annotation = 15;
optional string context_id = 16;
optional string system_session_id = 17;
optional bool requires_acknowledgement = 18;
optional string created_by = 19;
optional string origin_name = 20;
optional string origin_address = 21;
optional AcknowledgementDetailsPb acknowledgement_details = 22;
repeated UserIdPb recipients = 23;

message AcknowledgementDetailsPb {

    optional AcknowledgementTypePb acknowledgement_type = 1;
    optional string ack_time = 2;
    optional string ack_user = 3;

    enum AcknowledgementTypePb{
        POSITIVE = 0;
        NEGATIVE = 1;
    }
}

message UserIdPb {

    optional string user_id = 1;
}

enum SeverityPb{
    ALARM = 0;
    WARNING = 1;
    INFO = 2;
    DEBUG = 3;
}

enum MessageTypePb{
    LOG = 0;
    OPERATIONAL = 1;
    USER = 2;
    TRACE = 3;
}
}
```

6.7.3.3.15 EGSCC Parameter TM Definitions

Protobuf format is used to deliver EGSCC Parameter TM definitions data efficiently. Binary is one of the available formats EGSCC Parameter TM Definitions request submitters can choose from.

6.7.3.3.15.1 Data

Following is the GPB `EgscParamDefinitions` message definition:

```
message EgscParamDefinitionRecord
{
    optional string name = 1;
    optional string description = 2;
    optional string unit = 3;
    optional string type = 4;
```

```
    optional bool active = 5;
    optional string uuid = 6;
}
message EgsgccParamDefinitions
{
    repeated EgsgccParamDefinitionRecord entry = 1;
}
```

6.7.3.3.16 EGSCC Out of Limit Record Report

The Protobuf format is used to deliver OOL report data from EGSCC efficiently. Binary is one of the available formats that the OOL Report request can choose from.

6.7.3.3.16.1 Data

Following is the GPB `EgsgccOolReport` message definition:

```
message EgsgccOolReportBinary {
    optional OolObjectStatePb ool_object_state = 1;
    optional ActionRequiredPb action_required = 2;
    repeated CheckResultPb check_results = 3;
    optional GlobalStatePb global_state = 4;
    optional bool is_full_state = 5;
    optional bool is_outdated = 6;
    optional OolObjectProcessingStatePb
ool_object_processing_state = 7;
    optional string reception_route = 8;
    optional StateTransitionDetailsPb state_transition_details
= 9;

    optional AlarmPb summary_alarm_state = 10;
    optional ValidityTypePb validity = 11;
    optional ParameterValuePb value = 12;
    optional bool value_changed = 13;
}

message OolObjectStatePb {
    optional string alias = 1;
    repeated CategoryMarkerPb categories = 2;
    optional DataPartitionIdPb data_partition_id = 3;
    optional string egsgcc_reception_time = 4;
    optional string generation_time = 5;
    optional GenerationTimeQualityPb generation_time_quality = 6;
    optional MonitoringUpdateTypePb monitoring_update_type = 7;
    optional string processing_time = 8;
    optional string mcm_path = 9;
    optional string definition_id = 10;
}

message CheckResultPb {
    optional AlarmStatePb alarm_state = 1;
    optional bool enabled = 2;
    optional bool forced = 3;
    optional string id = 4;
    optional int64 violated_samples_counter = 5;
}

message CategoryMarkerPb {
    optional bool last_update = 1;
```

```

    optional CategoryTypePb type = 2;
}

message DataPartitionIdPb {
    optional string dataspace_id = 1;
    optional string id = 2;
}

message ParameterValuePb {
    optional string update_time = 1;
    optional ValuePb value = 2;
}

message ValuePb {
    optional ValueDefinitionPb value_definition = 1;
    optional SimpleValuePb simple_value = 2;
    optional AggregatePb aggregate = 3;
    optional SequencePb sequence_value = 4;
    optional EnumValuePb enum_value = 5;
    optional ReferenceValuePb reference_value = 6;
    optional DeducedValuePb deduced_value = 7;
}

message ValueDefinitionPb {
    optional CompoundTypePb compound_type = 1;
    optional string name = 2;
    optional SimpleValueDefinitionPb simple_value_definition = 3;
    optional AggregateDefinitionPb aggregate_definition = 6;
    optional SequenceDefinitionPb sequence_value_definition = 7;
    optional EnumValueDefinitionPb enum_value_definition = 9;
    optional ReferenceValueDefinitionPb eference_value_definition =
10;
    optional DeducedValueDefinitionPb deduced_value_definition = 11;
}

message AlarmPb {
    optional string check_identifier = 1;
    optional string description = 2;
    optional AlarmStatePb alarm_state = 3;
}

enum CompoundTypePb {
    SIMPLE_VALUE = 0;
    OBJECT = 1;
    MATRIX = 2;
    AGGREGATE = 3;
    SEQUENCE = 4;
    ANY = 5;
    ENUM_VALUE = 6;
    REFERENCE = 7;
    DEDUCED = 8;
    SELECTION = 9;
}

message SimpleValueDefinitionPb {
    optional DataTypePb data_type = 1;
    optional string data_unit = 2;
}

enum DataTypePb {
    INT8 = 1;

```

```
    INT16 = 2;
    INT32 = 3;
    INT64 = 4;
    UINT8 = 5;
    UINT16 = 6;
    UINT32 = 7;
    UINT64 = 8;
    FLOAT = 9;
    DOUBLE = 10;
    CHAR = 11;
    WIDE_CHAR = 12;
    BOOL = 13;
    BIT_STRING = 14;
    ASCII_STRING = 15;
    UTF16_STRING = 16;
    ABSOLUTE_TIME = 17;
    DURATION = 18;
    INTERVAL = 19;
    INVALID = 20;
    BYTE_STRING = 21;
}

message AggregateDefinitionPb {
    optional int64 field_count = 1;
}

message DeducedValueDefinitionPb {
}

message EnumValueDefinitionPb {
}

message ReferenceValueDefinitionPb {
}

message SequenceDefinitionPb {
    optional ValueDefinitionPb element_type = 1;
}

message SimpleValuePb {
    optional double double_value = 1;
    optional float float_value = 2;
    optional int32 int32_value = 3;
    optional int64 int64_value = 4;
    optional uint32 uint32_value = 5;
    optional uint64 uint64_value = 6;
    optional sint32 sint32_value = 7;
    optional sint64 sint64_value = 8;
    optional fixed32 fixed32_value = 9;
    optional fixed64 fixed64_value = 10;
    optional sfixed32 sfixed32_value = 11;
    optional sfixed64 sfixed64_value = 12;
    optional bool bool_value = 13;
    optional string string_value = 14;
    optional bytes bytes_value = 15;
    optional bool is_engineering_value = 16;
}
```

```
message AggregatePb {
    repeated SimpleValuePb simple_value = 1;
}

message SequencePb {
    repeated SimpleValuePb simple_value = 1;
}

message EnumValuePb {
    optional int64 number = 1;
    optional string text = 2;
}

message ReferenceValuePb {
    optional McmObjectIdPb mcm_object_id = 1;
}

message DeducedValuePb {
    optional ReferenceValuePb reference_value = 1;
    optional SimpleValuePb simple_value = 2;
    optional EnumValuePb enum_value = 3;
}

message McmObjectIdPb {
    optional string definition_id = 1;
    optional int64 occurrence_id = 2;
}

message StateTransitionDetailsPb {
    optional ParameterValuePb last_nominal = 1;
    optional AlarmPb previous_summary_alarm_state = 2;
    optional ParameterValuePb previous_value = 3;
    optional bool state_changed = 4;
    optional ParameterValuePb state_transition_value = 5;
}

message AlarmPb {
    optional string check_identifier = 1;
    optional string description = 2;
    optional AlarmStatePb alarm_state = 3;
}

enum AlarmStatePb {
    NOT_CHECKED = 0;
    NOMINAL = 1;
    WARNING = 2;
    ALARM = 3;
    NOMINAL_NO_CHECK_DEFINED = 4;
    NOMINAL_CHECKS_DISABLED = 5;
    UNKNOWN_STATE = 6;
    VIOLATED = 7;
}

enum CategoryTypePb {
    ALL_ACTIVITIES = 0;
    SCHEDULED_ACTIVITIES = 1;
    GROUND_SCHEDULED_ACTIVITIES = 2;
    EXTERNALLY_SCHEDULED_ACTIVITIES = 3;
    RUNNING_ACTIVITIES = 4;
    FORCED_PARAMETERS = 5;
    INVALID_PARAMETERS = 6;
}
```



```
    ALARM_PARAMETERS = 7;
    ALL_EVENTS = 8;
}

enum GenerationTimeQualityPb {
    ACCURATE = 0;
    INACCURATE = 1;
    ESTIMATED = 2;
    BAD = 3;
    GROUND = 4;
}

enum MonitoringUpdateTypePb {
    CREATION = 0;
    DELETION = 1;
    UPDATE = 2;
    CHANGE = 3;
}

enum ActionRequiredPb {
    NO_ACTION_REQUIRED_DEFINED = 0;
    OUTSTANDING = 1;
    ACTION_TAKEN_OVER = 2;
    PREVIOUS_OUTSTANDING = 3;
    NO_ACTION_REQUIRED = 4;
}

enum AlarmStatePb {
    NOT_CHECKED = 0;
    NOMINAL = 1;
    WARNING = 2;
    ALARM = 3;
    NOMINAL_NO_CHECK_DEFINED = 4;
    NOMINAL_CHECKS_DISABLED = 5;
    UNKNOWN_STATE = 6;
    VIOLATED = 7;
}

enum GlobalStatePb {
    ALARM = 0;
    ALARM_ACTION_REQUIRED = 1;
    DISABLED = 2;
    ERROR = 3;
    FORCED = 4;
    INCONSISTENT_STATE = 5;
    INVALID = 6;
    NO_VALUE = 7;
    NOMINAL = 8;
    NOMINAL_CHECKS_DISABLED = 9;
    NOMINAL_DEFAULT_VALUE = 10;
    NOMINAL_NO_CHECK_DEFINED = 11;
    SUSPENDED = 12;
    UNKNOWN_STATE = 13;
    WARNING = 14;
    WARNING_ACTION_REQUIRED = 15;
}

enum OolObjectProcessingStatePb {
    PROCESSED = 0;
    FORCED = 1;
    DISABLED = 2;
}
```

```
SUSPENDED = 3;
NOT_RECEIVED = 4;
INVALID_DEFINITION = 5;
ERROR = 6;
}

enum ValidityTypePb {
    VALID = 0;
    UNKNOWN = 1;
    INVALID = 2;
    UNKNOWN_VALUE = 3;
}
```

6.8 *RawSourceBinary*

This format is the simplest format we currently have in EDDS and can only be requested for TM Report request type, both parc and dataprovion service. In practical terms, for TM packets, this is nothing more than a continuous data message of CCSDS TM source packets (without any additional header).

7. The output is a binary file of the continuous raw data as provided by RawBodyData attribute. Naming Conventions

7.1 Request ID Convention

The Request ID is formed using the information in the following table:

Field	Type	Possible Values
RequestType	String	BatchRequest, StreamRequest, Cancel, DeleteData, DeleteRequest, AccountRequest
Subtype	String	Taken from DataAccessDataElement (See Section D.23)
Mission	String	Any alphanumeric value
Domain	String	Typically, numbers from 0 – 51
Username	String	The user ID of the user performing the request
Time	String	Time of the creation of the request in DOY form in GMT
RandomStr	String	Random 4 char string to make Request ID unique

Table 28 - Request ID Convention

The Request ID can either be on of the following two formats:

[RequestType].[Username].[yyyy.DDD.hh.mm.ss.SSS]

Or

[RequestType].[Subtype].[Mission].[Domain].[yyyy.DDD.hh.mm.ss.SSS].[RandomStr]

The first form is used for changes to user and mission LDAP account information (AccountRequest types) and the other form is used for all other request types. The Request ID is created by the EDDS Archiver (by the ScheduleUtility class) when the request is added to the database.

7.2 Response ID Convention

The Response ID is formed using the information in the following table:

Field	Type	Possible Values
RequestId	RequestId	Same as Section 7
Time	String	Time the response file was created in DOY form in GMT
Counter	Int	The response file number, starting at 1.
Extension	String	The file type

Table 29 - Response ID Convention

The Response ID is in the following format:

[RequestType].[Subtype].[Mission].[Domain].[yyyy.DDD.hh.mm.ss.SSS]@[yyyy.DDD.hh.mm.ss.SSS].[Counter].[Extension]

7.2.1 EDDS Stream Client's response file name convention

7.2.1.1 *TM requests*

EDDS stream client will save the data in the files with the following naming convention for TM requests: **"TmRecording_"** + day, month, year, hour, minute, second, milli second of storing time + format extention. Where all fields are separated by a **"_"**.

For example: TmRecording_14_08_2017_17_27_34_2334.xml

7.2.1.2 *TC requests*

EDDS stream client will save the data in the files with the following naming convention for TC requests: **"TcRecording_"** + day, month, year, hour, minute, second, milli second of storing time + format extention. Where all fields are separated by a **"_"**.

For example: TcRecording_14_08_2017_17_27_34_2334.xml

7.2.1.3 *OOL requests*

EDDS stream client will save the data in the files with the following naming convention for OOL requests: **"OolRecording_"** + day, month, year, hour, minute, second, milli second of storing time + format extention. Where all fields are separated by a **"_"**.

For example: OolRecording_14_08_2017_17_27_34_2334.xml

7.2.1.4 *Event requests*

EDDS stream client will save the data in the files with the following naming convention for Event requests: **"EvRecording_"** + day, month, year, hour, minute, second, milli second of storing time + format extention. Where all fields are separated by a **"_"**.

For example: EvRecording_14_08_2017_17_27_34_2334.xml

7.2.1.5 *Param requests*

EDDS stream client will save the data in the files with the following naming convention for Param requests: **"ParamRecording_"** + day, month, year, hour, minute, second, milli second of storing time + format extention. Where all fields are separated by a **"_"**.

For example: ParamRecording_14_08_2017_17_27_34_2334.xml

8. Delivery

8.1 *Delivery Mechanism*

The EDDS supports the following delivery mechanisms for the delivery of data to the end user.

The batch request response data can be delivered to the user via File Server or EDDS Server each file separately as soon as EDDS Server is finished processing a file and it is available for Delivery Manager.

8.1.1 File Server

Data types that support the 'File Server' delivery mechanism is delivered over SFTP or FTP using the EDDS web server. Direct delivery requires the end user to be running a server that supports SFTP/FTP.

In order to user SFTP the user has to provide username and password of the remote file server. In case these data is not provided, the web server will automatically try to use FTP.

8.1.2 EDDS Server

Data types that support the EDDS Server delivery mechanism are stored on an EDDS server. Also, when the File Server delivery fails, the data is still stored in EDDS file server.

8.1.3 Stream

Data types that support Stream delivery require the end user to run a stream client application that can be downloaded from the EDDS web site and installed on the user's machine. Stream delivery supports online and offline data where, in the context of the EDDS, offline is taken to refer to the retrieval of previously archived data and online is taken to refer to the routing of data as it is received on the relevant control system. In addition, online data delivery supports two modes of operation: timely and complete. Timely mode ensures data is delivered within a specified time and will drop data if this criterion cannot be met. Complete mode ensures all data is delivered, although there may be considerable delay in the arrival of the complete data set.

It is possible to use EDDS stream client to save streaming data in files, read chapter 4.3.4 for more information.

9. Web Services Description

The EDDS uses web services as the fundamental mechanism to provide the functionality behind EDDS batch services and EDDS stream services. This section of the EUICD details the structure of the EDDS Web Services.

9.1 *Web Service Introduction*

The EDDS provides access to data via Web Services that are defined using Web Services Description Language (WSDL) see [RD-3] and REST APIs. The Web Services described in this section of the document allows client applications access to batch and user management services.

The services provide operations that allow a client to be authenticated, make requests for batch data, make requests for stream data, perform user management and receive data in response to those requests.

A WSDL service description is an XML document that contains a set of definitions for:

- Port Types
- Messages
- Bindings
- Services

A port type is a named set of abstract operations and the abstract messages involved. Message definitions are always considered to be an abstract definition of the message content. A message binding describes how the abstract content is mapped into a concrete format. However, in the case of the EDDS the content is intended to map directly onto the XML schemas. The bindings provide the protocol details for operations and messages defined by a particular port type. In the case of the EDDS, the bindings are to a SOAP protocol using XML messages. Finally, a service definition groups a set of related ports together. The definition of the EDDS Web Services is defined in the XML documents,

Port operations defined by EDDS provide a means to indicate if an error has occurred during the request of an operation.) An exception is raised when a service request cannot be executed with success.

The WSDL file can be viewed in a web browser by simply entering the end point URL normally passed to the client application into the web browser's address bar.

A REST API service description is composed by:

- Endpoint
- Input
- Output

All the information is exchanged using JSON format. Sometimes, for input and output are used complex types. In order to use endpoints that accept in input a complex type, you must first translate the complex type to JSON format and the submit it to the endpoint. The returning complex types are automatically translated to JSON format.

9.2 *PortTypes*

A port type is a named set of abstract operations and the abstract messages involved. The port type name attribute provides a unique name among all port types defined within in the enclosing service definition document. WSDL has four transmission primitives that an endpoint can support:

- One-way - The endpoint receives a message.
- Request-response - The endpoint receives a message, and sends a correlated message.
- Solicit-response - The endpoint sends a message, and receives a correlated message.

- Notification - The endpoint sends a message.

WSDL refers to these primitives as operations.

The operation instances defined in the EDDS port types are based on ‘one-way’ and ‘request-response’ transmission primitives. The following sections describe the port types that have been defined for the EDDS.

9.2.1 EDDS Port Type

The EDDS port type provides the operations that allow the authentication of a user and subsequent allocation of a session identifier. All users of the EDDS are assigned a session and it is under this session that a series of operations are made. A session will be timed out by the EDDS server if inactive, however it is strongly recommended that applications should actively close the session when appropriate. This ensures that server resources are freed as soon as possible to service new session requests. Not following this guideline will result in the potential delaying (or rejection) of session requests. The timeout period of sessions can be changed by the EDDS Administrator in the EDDS Web Server configuration in the file “web.xml”. The authentication port type also provides operations for the client to query the roles and missions available within the session.

The EDDS port type is a component that is used by both batch and stream services.

It provides the set of operations available for the request and receipt of data (and acknowledgements) within the scope of an EDDS batch service.

It provides the set of operations available for the execution of user management requests.

9.2.1.1 EddsPortType

The EDDS port type describes the named set of messages that a service sends and/or receives

operations	batchRequest <i>The batch request operation</i> input ws:batchRequest output ws:batchRequestResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> streamRequest <i>The stream request operation</i> input ws:streamRequest output ws:streamRequestResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i>
------------	---

	fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getStatus <i>The get status operation</i> input ws:getStatus output ws:getStatusResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> reSubmit <i>The re-submit operation</i> input ws:reSubmit output ws:reSubmitResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getStatuses <i>Gets the status of a list of jobs</i> input ws:getStatuses
--	--

	output ws:getStatusesResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getResponse <i>The get response operation</i> input ws:getResponse output ws:getResponseResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getRequest <i>The get request operation</i> input ws:getRequest output ws:getRequestResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i>
--	---

	input ws:getParamDefinitions output ws:getParamDefinitionsResponse fault ws:requestFault An error occurs during the elaboration of the request fault ws:authorizationFault If the user has no privileges to perform the request fault ws:sessionFault If the user is not logged in getCatalogue Gets the catalogue for the specified data source input ws:getCatalogue output ws:getCatalogueResponse fault ws:requestFault An error occurs during the elaboration of the request fault ws:authorizationFault If the user has no privileges to perform the request fault ws:sessionFault If the user is not logged in getFarcCatalogue Gets the FARC catalogue for the specified mission and domain @deprecated use getCatalogue instead input ws:getFarcCatalogue output ws:getFarcCatalogueResponse fault ws:requestFault An error occurs during the elaboration of the request fault ws:authorizationFault If the user has no privileges to perform the request fault ws:sessionFault If the user is not logged in
--	--

	getDataspaces <i>Gets the dataspaces for the specified mission and request type</i> input ws:getDataspaces output ws:getDataspacesResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no privileges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getAlias <i>Gets aliases for the specified mission</i> input ws:getAlias output ws:getAliasResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getLastConsolidation <i>Gets the last consolidation time for the specified mission, request type and dataspace name if given.</i> input ws:getLastConsolidation output ws:getLastConsolidationResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault
--	--

	<i>If the user is not logged in</i> getAllowedRequestTypes <i>Gets the request types that the currently logged-in user has permission to issue.</i> input <u>ws:getAllowedRequestTypes</u> output <u>ws:getAllowedRequestTypesResponse</u> fault <u>ws:requestFault</u> <i>An error occurs during the elaboration of the request</i> fault <u>ws:authorizationFault</u> <i>If the user has no priviledges to perform the request</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> cancel <i>The cancel operation</i> input <u>ws:cancel</u> output <u>ws:cancelResponse</u> fault <u>ws:requestFault</u> <i>An error occurs during the elaboration of the request</i> fault <u>ws:authorizationFault</u> <i>If the user has no priviledges to perform the request</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> suspend <i>The suspend operation</i> input <u>ws:suspend</u> output <u>ws:suspendResponse</u> fault <u>ws:requestFault</u> <i>An error occurs during the elaboration of the request</i> fault <u>ws:authorizationFault</u> <i>If the user has no priviledges to perform the request</i>
--	--

	fault ws:sessionFault <i>If the user is not logged in</i>
	resume <i>The resume operation</i> input ws:resume output ws:resumeResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i>
	fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i>
	fault ws:sessionFault <i>If the user is not logged in</i>
	deleteData <i>The operation to delete response data</i> input ws:deleteData output ws:deleteDataResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i>
	fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i>
	fault ws:sessionFault <i>If the user is not logged in</i>
	deleteRequest <i>The operation to delete a request</i> input ws:deleteRequest output ws:deleteRequestResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i>

	fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getJobs <i>The get jobs operation</i> input ws:getJobs output ws:getJobsResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getJobsWithStatusAndLimit input ws:getJobsWithStatusAndLimit output ws:getJobsWithStatusAndLimitResponse fault ws:authorizationFault <i>If the user has no privileges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getJobsWithStatus <i>Gets the jobs for the specified start and end date. The message contains a java.util.ArrayList of type RequestInfoRec encoded in Base64</i> input ws:getJobs output ws:getJobsWithStatusResponse
--	---

	fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getHistoricalLog <i>Gets the historical system logs</i> input ws:getHistoricalLog output ws:getHistoricalLogResponse fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> stopHistoricalLogRetrieval <i>Stops the retrieval of historical log messages</i> input ws:stopHistoricalLogRetrieval output ws:stopHistoricalLogRetrieval Response fault ws:requestFault <i>An error occurs during the elaboration of the request</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getTransformations
--	---

	<i>Gets the available XSL Transformations for the given mission and request type</i> input <u>ws:getTransformations</u> output <u>ws:getTransformationsResponse</u> fault <u>ws:requestFault</u> <i>An error occurs during the elaboration of the request</i> fault <u>ws:authorizationFault</u> <i>If the user has no privileges to perform the request</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> logIn <i>Authenticate the user and start the session</i> input <u>ws:logIn</u> <i>The credentials of the user: username and password</i> output <u>ws:logInResponse</u> <i>No response given</i> fault <u>ws:authenticationFault</u> <i>If the username or password are incorrect or the LDAP server could not be reached</i> fault <u>ws:passwordExpiredFault</u> <i>If the user's password has expired and needs to be changed</i> hasSessionExpired <i>Checks if the user's session has expired, and return true if it has. This operation has a one-way transport.</i> input <u>ws:checkSession</u> <i>Input is not used. JAX-WS doesn't support notification style messages.</i> output <u>ws:checkSessionResponse</u> <i>Returns true if the session has expired or has not been started</i> logOut
--	--

	<i>Close the session for the user. This operation has a one-way transport</i> input <u>ws:logOut</u> checkUserPassword <i>Check if the password supplied matches the user's current password. Used to double check the user's current password before allowing the user to change their password.</i> input <u>ws:checkUserPassword</u> <i>The user's current password in plain text</i> output <u>ws:checkUserPasswordResponse</u> <i>No output given</i> fault <u>ws:authenticationFault</u> <i>If the password doesn't match or the LDAP server could not be reached</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> accountRequest <i>The account request operation</i> input <u>ws:accountRequest</u> <i>The actual request</i> output <u>ws:accountRequestResponse</u> fault <u>ws:authorizationFault</u> <i>If the user has no priviledges to perform the request</i> fault <u>ws:requestFault</u> <i>An error occurs in the elaboration of the request</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> getUsers <i>Retrives the list of users</i> input <u>ws:getUsers</u>
--	--

	<i>Flag to indicate whether to fetch all the users</i> output ws:getUsersResponse <i>The list of users</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getMissions <i>the list of missions</i> input ws:getMissions <i>Empty message</i> output ws:getMissionsResponse <i>The list of missions</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getMissionsAndDomains <i>the list of missions with their domains</i> input ws:getMissionsAndDomains <i>Empty message</i>
--	--

	output ws:getMissionsAndDomainsResponse <i>The list of missions</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getRoles <i>Returns the roles the user has permission to use for the specified mission</i> input ws:getRoles <i>The name of the mission</i> output ws:getRolesResponse <i>The role details</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getUserAccountDetails <i>Returns the details for the specified user</i> input ws:getUserAccountDetails <i>The name of the user</i> output ws:getUserAccountDetailsResponse
--	--

	<i>The user's details</i> fault <u>ws:authorizationFault</u> <i>If the user has no priviledges to perform the request</i> fault <u>ws:requestFault</u> <i>An error occurs in the elaboration of the request</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> getMissionDetails <i>Returns the details for the specified mission</i> input <u>ws:getMissionDetails</u> <i>The name of the mission</i> output <u>ws:getMissionDetailsResponse</u> <i>The mission details</i> fault <u>ws:authorizationFault</u> <i>If the user has no priviledges to perform the request</i> fault <u>ws:requestFault</u> <i>An error occurs in the elaboration of the request</i> fault <u>ws:sessionFault</u> <i>If the user is not logged in</i> getUserQuota <i>Returns the quota usage for the specified mission and logged on user</i> input <u>ws:getUserQuota</u> <i>The name of the mission</i> output <u>ws:getUserQuotaResponse</u> <i>The quota details for the mission and logged on user</i>
--	---

	fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> getQuotaDetails <i>Returns the quota usage for the specified mission and logged on user.</i> input ws:getQuotaDetails <i>The name of the mission</i> output ws:getQuotaDetailsResponse <i>The quota details for the mission and user</i> fault ws:authorizationFault <i>If the user has no priviledges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i> hasPermissionToSeeAllUsers <i>Checks if user has privileges to see all users data.</i> input ws:hasPermissionToSeeAllUsers <i>Username</i> output ws:permissionToSeeAllUsersResponse <i>Returns true if the user has permission to see other users data</i> fault ws:authorizationFault
--	--

	<i>If the user has no privileges to perform the request</i> fault ws:requestFault <i>An error occurs in the elaboration of the request</i> fault ws:sessionFault <i>If the user is not logged in</i>
used by	binding EddsBinding

9.3 Messages

This section describes the XML messages that EDDS sends and receive in order to provide services.

9.3.1 message accountRequest

The message contains the details of a account management request

parts	accountRequest element model:AccountRequestMessagePart
used by	Operation accountRequest in PortType EddsPortType
source	<wsdl:message name="accountRequest"> <wsdl:documentation>The message contains the details of a account management request</wsdl:documentation> <wsdl:part name="accountRequest" element="model:AccountRequestMessagePart"/> </wsdl:message>

9.3.2 message accountRequestResponse

Empty message

parts	
used by	Operation accountRequest in PortType EddsPortType
source	<wsdl:message name="accountRequestResponse"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message>

9.3.3 message authenticationFault

The message contains details related to a authentication fault

parts	authenticationFault element model:AuthenticationFault
used by	Operation logIn in PortType EddsPortType

	Operation checkUserPassword in PortType EddsPortType
source	<code><wsdl:message name="authenticationFault"></code> <code><wsdl:documentation>The message contains details related to a authentication</code> <code>fault</wsdl:documentation></code> <code><wsdl:part name="authenticationFault" element="model:AuthenticationFault"/></code> <code></wsdl:message></code>

9.3.4 message authorizationFault

The message contains details related to a authorization fault

parts	authorizationFault element model:AuthorizationFault
used by	Operation batchRequest in PortType EddsPortType Operation streamRequest in PortType EddsPortType Operation getStatus in PortType EddsPortType Operation getStatues in PortType EddsPortType Operation getResponse in PortType EddsPortType Operation getRequest in PortType EddsPortType Operation getStreamRequest in PortType EddsPortType Operation startStreamData in PortType EddsPortType Operation getParamDefinitions in PortType EddsPortType Operation getFarCatalogue in PortType EddsPortType Operation cancel in PortType EddsPortType Operation suspend in PortType EddsPortType Operation resume in PortType EddsPortType Operation deleteData in PortType EddsPortType Operation deleteRequest in PortType EddsPortType Operation getJobs in PortType EddsPortType Operation getJobsWithStatus in PortType EddsPortType Operation getJobsWithStatusAndLimit in PortType EddsPortType Operation getHistoricalLog in PortType EddsPortType Operation stopHistoricalLogRetrieval in PortType EddsPortType Operation getTransformations in PortType EddsPortType Operation accountRequest in PortType EddsPortType Operation getUsers in PortType EddsPortType Operation getMissions in PortType EddsPortType Operation getMissionsAndDomains in PortType EddsPortType Operation getRoles in PortType EddsPortType Operation getUserAccountDetails in PortType EddsPortType Operation getMissionDetails in PortType EddsPortType Operation getUserQuota in PortType EddsPortType Operation hasPermissionToSeeAllUsers in PortType EddsPortType
source	<code><wsdl:message name="authorizationFault"></code> <code><wsdl:documentation>The message contains details related to a authorization</code> <code>fault</wsdl:documentation></code> <code><wsdl:part name="authorizationFault" element="model:AuthorizationFault"/></code> <code></wsdl:message></code>

9.3.5 message batchRequest

The message contains the details of a batch request

parts	batchRequest element model:RequestMessagePart
used by	Operation batchRequest in PortType EddsPortType
source	<code><wsdl:message name="batchRequest"></code> <code><wsdl:documentation></code> The message contains the details of a batch request <code></wsdl:documentation></code> <code><wsdl:part name="batchRequest" element="model:RequestMessagePart"/></code> <code></wsdl:message></code>

9.3.6 message batchRequestResponse

The message contains job ID associated to a batch request

parts	batchRequestResponse element model:JobListIdPart
used by	Operation batchRequest in PortType EddsPortType
source	<code><wsdl:message name="batchRequestResponse"></code> <code><wsdl:documentation></code> The message contains job ID associated to a batch request <code></wsdl:documentation></code> <code><wsdl:part name="batchRequestResponse" element="model:JobListIdPart"/></code> <code></wsdl:message></code>

9.3.7 message cancel

The message contains the details of "cancel" requests (the job IDs)

parts	cancel element model:CancelPartList
used by	Operation cancel in PortType EddsPortType
source	<code><wsdl:message name="cancel"></code> <code><wsdl:documentation></code> The message contains the details of "cancel" requests (the job IDs) <code></wsdl:documentation></code> <code><wsdl:part name="cancel" element="model:CancelPartList"/></code> <code></wsdl:message></code>

9.3.8 message suspend

The message contains the details of "suspend" request (the job IDs)

parts	suspend element model:SuspendPartList
used by	Operation suspend in PortType EddsPortType
source	<code><wsdl:message name="suspend"></code> <code><wsdl:documentation></code> The message contains the details of "suspend" requests (the job IDs) <code></wsdl:documentation></code> <code><wsdl:part name="suspend" element="model:SuspendPartList"/></code> <code></wsdl:message></code>

9.3.9 message resume

The message contains the details of "resume" requests (the job IDs)

parts	resume element model:ResumePartList
used by	Operation resume in PortType EddsPortType
source	<code><wsdl:message name="resume"> <wsdl:documentation>The message contains the details of "resume" requests (the job IDs)</wsdl:documentation> <wsdl:part name="resume" element="model:ResumePartList"/> </wsdl:message></code>

9.3.10 message cancelResponse

The message contains the job IDs associated with the cancel requests

parts	cancelResponse element model:AcknowledgementPartList
used by	Operation cancel in PortType EddsPortType
source	<code><wsdl:message name="cancelResponse"> <wsdl:documentation>The message contains the job IDs associated with the cancel requests</wsdl:documentation> <wsdl:part name="cancelResponse" element="model:AcknowledgementPartList"/> </wsdl:message></code>

9.3.11 message suspendResponse

The message contains the job IDs associated with the suspend requests

parts	suspendResponse element model:AcknowledgementPartList
used by	Operation suspend in PortType EddsPortType
source	<code><wsdl:message name="suspendResponse"> <wsdl:documentation>The message contains the job IDs associated with the suspend requests</wsdl:documentation> <wsdl:part name="suspendResponse" element="model:AcknowledgementPartList"/> </wsdl:message></code>

9.3.12 message resumeResponse

The message contains the job IDs associated with the resume requests

parts	resumeResponse element model:AcknowledgementPartList
used by	Operation resume in PortType EddsPortType
source	<code><wsdl:message name="resumeResponse"></code>

	<wsdl:documentation>The message contains the job IDs associated with the resume requests</wsdl:documentation> <wsdl:part name="resumeResponse" element="model:AcknowledgementPartList"/> </wsdl:message>
--	---

9.3.13 message checkSession

Empty message

parts	
used by	Operation hasSessionExpired in PortType EddsPortType
source	<wsdl:message name="checkSession"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message>

9.3.14 message checkSessionResponse

True if session has expired (or not started)

parts	sessionResponse element model:SessionExpired
used by	Operation hasSessionExpired in PortType EddsPortType
source	<wsdl:message name="checkSessionResponse"> <wsdl:documentation>True if session has expired (or not started)</wsdl:documentation> <wsdl:part name="sessionResponse" element="model:SessionExpired"/> </wsdl:message>

9.3.15 message checkUserPassword

The message contains the password to be verified

parts	checkUserPassword element model:Password
used by	Operation checkUserPassword in PortType EddsPortType
source	<wsdl:message name="checkUserPassword"> <wsdl:documentation>The message contains the password to be verified</wsdl:documentation> <wsdl:part name="checkUserPassword" element="model:Password"/> </wsdl:message>

9.3.16 message checkUserPasswordResponse

Empty message

parts	
-------	--

used by	Operation checkUserPassword in PortType EddsPortType
source	<code><wsdl:message name="checkUserPasswordResponse"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message></code>

9.3.17 message deleteData

The message contains the details of a request to delete response data (a job ID)

parts	deleteData element model:DeleteDataPart
used by	Operation deleteData in PortType EddsPortType
source	<code><wsdl:message name="deleteData"> <wsdl:documentation>The message contains the details of a request to delete response data (a job ID)</wsdl:documentation> <wsdl:part name="deleteData" element="model:DeleteDataPart"/> </wsdl:message></code>

9.3.18 message deleteDataResponse

The message contains the job IDs associated to a request to delete response data

parts	deleteDataResponse element model:AcknowledgementPartList
used by	Operation deleteData in PortType EddsPortType
source	<code><wsdl:message name="deleteDataResponse"> <wsdl:documentation>The message contains the job IDs associated to a request to delete response data</wsdl:documentation> <wsdl:part name="deleteDataResponse" element="model:AcknowledgementPartList"/> </wsdl:message></code>

9.3.19 message deleteRequest

The message contains the details of a request to delete requests (the job IDs)

parts	deleteRequest element model>DeleteRequestPart
used by	Operation deleteRequest in PortType EddsPortType
source	<code><wsdl:message name="deleteRequest"> <wsdl:documentation>The message contains the details of a request to delete requests (the job IDs)</wsdl:documentation> <wsdl:part name="deleteRequest" element="model>DeleteRequestPart"/> </wsdl:message></code>

9.3.20 message deleteRequestResponse

The message contains the job IDs associated to the deletion requests

parts	deleteRequestResponse element model:AcknowledgementPartList
used by	Operation deleteRequest in PortType EddsPortType
source	<code><wsdl:message name="deleteRequestResponse"> <wsdl:documentation>The message contains the job IDs associated to the deletion requests</wsdl:documentation> <wsdl:part name="deleteRequestResponse" element="model:AcknowledgementPartList"/> </wsdl:message></code>

9.3.21 message getAllowedRequestTypes

Gets the request types the logged-in user has permission to issue

parts	
used by	Operation getAllowedRequestTypes in PortType EddsPortType
source	<code><wsdl:message name="getAllowedRequestTypes"> <wsdl:documentation> Gets the request types the logged-in user has permission to issue </wsdl:documentation> </wsdl:message></code>

9.3.22 message getAllowedRequestTypesResponse

Returns the list of request types the logged-in user has permission to issue

parts	allowedRequestTypesResponse element model:AllowedRequestTypes
used by	Operation getAllowedRequestTypes in PortType EddsPortType
source	<code><wsdl:message name="getAllowedRequestTypesResponse"> <wsdl:documentation>Returns the list of request types the logged-in user has permission to issue</wsdl:documentation> <wsdl:part name="allowedRequestTypesResponse" element="model:AllowedRequestTypes"/> </wsdl:message></code>

9.3.23 message getCatalogue

Gets the catalogue for the specified data source.

parts	getCatalogue element model:CatalogueRequest
used by	Operation getCatalogue in PortType EddsPortType
source	<code><wsdl:message name="getCatalogue"> <wsdl:documentation>Gets the catalogue for the specified data source.</wsdl:documentation> <wsdl:part name="getCatalogue" element="model:CatalogueRequest"/> </wsdl:message></code>

9.3.24 message getCatalogueResponse

Returns the allocated request ID for the request. This is used to take the asynchronous messages off the topic.

parts	catalogueResponse element model:JobIdPart
used by	Operation getCatalogue in PortType EddsPortType
source	<code><wsdl:message name="getCatalogueResponse"> <wsdl:documentation>Returns the allocated request ID for the request. This is used to take the asynchronous messages off the topic.</wsdl:documentation> <wsdl:part name="catalogueResponse" element="model:JobIdPart"/> </wsdl:message></code>

9.3.25 message getDataspaces

Gets the list of available dataspaces. For DARC, the first entry in the list is the currently active one.

parts	getDataspaces element model:DataspacesRequest
used by	Operation getDataspaces in PortType EddsPortType
source	<code><wsdl:message name="getDataspaces"> <wsdl:documentation>Gets the list of available dataspaces. For DARC, the first entry in the list is the currently active one.</wsdl:documentation> <wsdl:part name="getDataspaces" element="model:DataspacesRequest"/> </wsdl:message></code>

9.3.26 message getDataspacesResponse

Returns the list of available dataspaces. For DARC, the first entry in the list is the currently active one

parts	dataspacesResponse element model:ListString
used by	Operation getDataspaces in PortType EddsPortType
source	<code><wsdl:message name="getDataspacesResponse"> <wsdl:documentation>Returns the list of available dataspaces. For DARC, the first entry in the list is the currently active one</wsdl:documentation> <wsdl:part name="dataspacesResponse" element="model:ListString"/> </wsdl:message></code>

9.3.27 message getFarcCatalogue

Gets the FARC catalogue for the specified mission and domain (-1 represents the "No domain").
@deprecated use getCatalogue instead

parts	getFarcCatalogue element model:FarcCatalogueRequest
used by	Operation getFarcCatalogue in PortType EddsPortType

source	<wsdl:message name="getFarcCatalogue"> <wsdl:documentation>Gets the FARC catalogue for the specified mission and domain (-1 represents the "No domain"). @deprecated use getCatalogue instead</wsdl:documentation> <wsdl:part name="getFarcCatalogue" element="model:FarcCatalogueRequest"/> </wsdl:message>
--------	---

9.3.28 message getFarcCatalogueResponse

Returns the allocated request ID for the request. This is used to take the messages off the topic.
@deprecated

parts	farcCatalogueResponse element model:JobIdPart
used by	Operation getFarcCatalogue in PortType EddsPortType
source	<wsdl:message name="getFarcCatalogueResponse"> <wsdl:documentation>Returns the allocated request ID for the request. This is used to take the messages off the topic. @deprecated</wsdl:documentation> <wsdl:part name="farcCatalogueResponse" element="model:JobIdPart"/> </wsdl:message>

9.3.29 message getHistoricalLog

The message contains the start and end date to retrieve the historical log from EDDS

parts	getHistoricalLog element model:TimeWindow
used by	Operation getHistoricalLog in PortType EddsPortType
source	<wsdl:message name="getHistoricalLog"> <wsdl:documentation>The message contains the start and end date to retrieve the historical log from EDDS</wsdl:documentation> <wsdl:part name="getHistoricalLog" element="model:TimeWindow"/> </wsdl:message>

9.3.30 message getHistoricalLogResponse

Returns the allocated request ID for the request. This is used to take the messages off the topic.

parts	getHistoricalLogResponse element model:JobIdPart
used by	Operation getHistoricalLog in PortType EddsPortType
source	<wsdl:message name="getHistoricalLogResponse"> <wsdl:documentation>Returns the allocated request ID for the request. This is used to take the messages off the topic.</wsdl:documentation> <wsdl:part name="getHistoricalLogResponse" element="model:JobIdPart"/> </wsdl:message>

9.3.31 message getJobs

The message contains the start and end date to retrieve the jobs from EDDS

parts	getJobs element model:TimeWindow
used by	Operation getJobs in PortType EddsPortType Operation getJobsWithStatus in PortType EddsPortType
source	<code><wsdl:message name="getJobs"></code> <code><wsdl:documentation></code> The message contains the start and end date to retrieve the jobs from EDDS <code></wsdl:documentation></code> <code><wsdl:part name="getJobs" element="model:TimeWindow"/></code> <code></wsdl:message></code>

9.3.32 message getJobsResponse

The list of jobs

parts	getJobsResponse element model:JobListIdPart
used by	Operation getJobs in PortType EddsPortType
source	<code><wsdl:message name="getJobsResponse"></code> <code><wsdl:documentation></code> The list of jobs <code></wsdl:documentation></code> <code><wsdl:part name="getJobsResponse" element="model:JobListIdPart"/></code> <code></wsdl:message></code>

9.3.33 message getJobsWithStatus

The message contains the start and end date to retrieve the jobs from EDDS

parts	getJobsWithStatus element model:TimeWindow
source	<code><wsdl:message name="getJobsWithStatus"></code> <code><wsdl:documentation></code> The message contains the start and end date to retrieve the jobs from EDDS <code></wsdl:documentation></code> <code><wsdl:part name="getJobsWithStatus" element="model:TimeWindow"/></code> <code></wsdl:message></code>

9.3.34 message getJobsWithStatusAndLimit

parts	getJobsWithStatusAndLimit element model:TimeWindowWithLimit
used by	Operation getJobsWithStatusAndLimit in PortType EddsPortType
source	<code><wsdl:message name="getJobsWithStatusAndLimit"></code> <code><wsdl:part name="getJobsWithStatusAndLimit" element="model:TimeWindowWithLimit"/></code> <code></wsdl:message></code>

9.3.35 message getJobsWithStatusAndLimitResponse

parts	jobIdPart element model:JobIdPart
used by	Operation getJobsWithStatusAndLimit in PortType EddsPortType
source	<code><wsdl:message name="getJobsWithStatusAndLimitResponse"> <wsdl:part name="jobIdPart" element="model:JobIdPart"/> </wsdl:message></code>

9.3.36 message getJobsWithStatusResponse

Returns the allocated unique ID for the request. This is used to take the messages off the topic. The ID is a random string, not a Request ID.

parts	getJobsWithStatusResponse element model:JobIdPart
used by	Operation getJobsWithStatus in PortType EddsPortType
source	<code><wsdl:message name="getJobsWithStatusResponse"> <wsdl:documentation>Returns the allocated unique ID for the request. This is used to take the messages off the topic. The ID is a random string, not a Request ID.</wsdl:documentation> <wsdl:part name="getJobsWithStatusResponse" element="model:JobIdPart"/> </wsdl:message></code>

9.3.37 message getLastConsolidation

Gets the last consolidation time for the specified mission, request type and dataspace name if given.

parts	getLastConsolidation element model:LastConsolidationRequest
used by	Operation getLastConsolidation in PortType EddsPortType
source	<code><wsdl:message name="getLastConsolidation"> <wsdl:documentation>Gets the list of available dataspaces. For DARC, the first entry in the list is the currently active one.</wsdl:documentation> <wsdl:part name="getLastConsolidation" element="model:LastConsolidationRequest"/> </wsdl:message></code>

9.3.38 message getLastConsolidationResponse

Gets the last consolidation time for the specified mission, request type and dataspace name if given.

parts	lastConsolidationResponse element model:Long
used by	Operation getLastConsolidation in PortType EddsPortType
source	<code><wsdl:message name="getLastConsolidationResponse"> <wsdl:documentation>Returns the list of available dataspaces. For DARC, the first entry in the list is the currently active one</wsdl:documentation> <wsdl:part name="lastConsolidationResponse" element="model:Long"/> </wsdl:message></code>

9.3.39 message getMissionDetails

The message contains a mission's name

parts	getMissionDetails element model:MissionName
used by	Operation getMissionDetails in PortType EddsPortType
source	<code><wsdl:message name="getMissionDetails"> <wsdl:documentation>The message contains a mission's name</wsdl:documentation> <wsdl:part name="getMissionDetails" element="model:MissionName"/> </wsdl:message></code>

9.3.40 message getMissionDetailsResponse

The message contains the missions details associated to the mission's name as defined in the LDAP

parts	getMissionDetailsResponse element model:MissionDetails
used by	Operation getMissionDetails in PortType EddsPortType
source	<code><wsdl:message name="getMissionDetailsResponse"> <wsdl:documentation>The message contains the missions details associated to the mission's name as defined in the LDAP</wsdl:documentation> <wsdl:part name="getMissionDetailsResponse" element="model:MissionDetails"/> </wsdl:message></code>

9.3.41 message getMissions

Empty message

parts	
used by	Operation getMissions in PortType EddsPortType
source	<code><wsdl:message name="getMissions"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message></code>

9.3.42 message getMissionsResponse

The list of missions

parts	getMissionsResponse element model:MissionsList
used by	Operation getMissions in PortType EddsPortType
source	<code><wsdl:message name="getMissionsResponse"> <wsdl:documentation>The list of missions</wsdl:documentation> <wsdl:part name="getMissionsResponse" element="model:MissionsList"/> </wsdl:message></code>

9.3.43 message getMissionsAndDomains

Empty message

parts	
used by	Operation getMissionsAndDomains in PortType EddsPortType
source	<code><wsdl:message name="getMissionsAndDomains"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message></code>

9.3.44 message getMissionsAndDomainsResponse

The list of missions with their domains

parts	getMissionsAndDomainsResponse element model:MissionsDomainList
used by	Operation getMissionsAndDomains in PortType EddsPortType
source	<code><wsdl:message name="getMissionsAndDomainsResponse"> <wsdl:documentation>The list of missions with their domains</wsdl:documentation> <wsdl:part name="getMissionsAndDomainsResponse" element="model:MissionsDomainList"/> </wsdl:message></code>

9.3.45 message getParamDefinitions

Gets the parameter definitions for the specified mission.

parts	getParamDefinitions element model:MissionName
used by	Operation getParamDefinitions in PortType EddsPortType
source	<code><wsdl:message name="getParamDefinitions"> <wsdl:documentation>Gets the parameter definitions for the specified mission.</wsdl:documentation> <wsdl:part name="getParamDefinitions" element="model:MissionName"/> </wsdl:message></code>

9.3.46 message getParamDefinitionsResponse

Returns the allocated request ID for the request. This is used to take the messages off the topic.

parts	paramDefinitionsResponse element model:JobIdPart
used by	Operation getParamDefinitions in PortType EddsPortType
source	<code><wsdl:message name="getParamDefinitionsResponse"> <wsdl:documentation>Returns the allocated request ID for the request. This is used to take the messages off the topic.</wsdl:documentation> <wsdl:part name="paramDefinitionsResponse" element="model:JobIdPart"/></code>

	</wsdl:message>
--	-----------------

9.3.47 message getQuotaDetails

Get the quota details for the specified user and mission

parts	getQuotaDetailsMission element model:UserMission
used by	Operation getQuotaDetails in PortType EddsPortType
source	<wsdl:message name="getQuotaDetails"> <wsdl:documentation>Get the quota details for the specified user and mission</wsdl:documentation> <wsdl:part name="getQuotaDetailsMission" element="model:UserMission"/> </wsdl:message>

9.3.48 message getQuotaDetailsResponse

The message contains the quota usage details

parts	getQuotaDetailsResponse element model:UserQuotaDetails
used by	Operation getQuotaDetails in PortType EddsPortType
source	<wsdl:message name="getQuotaDetailsResponse"> <wsdl:documentation>The message contains the quota usage details</wsdl:documentation> <wsdl:part name="getQuotaDetailsResponse" element="model:UserQuotaDetails"/> </wsdl:message>

9.3.49 message getRequest

The message contains the details of a "request" request (a job ID)

parts	getRequest element model:JobIdPart
used by	Operation getRequest in PortType EddsPortType
source	<wsdl:message name="getRequest"> <wsdl:documentation>The message contains the details of a "request" request (a job ID)</wsdl:documentation> <wsdl:part name="getRequest" element="model:JobIdPart"/> </wsdl:message>

9.3.50 message getRequestResponse

The message contains the original request associated to the given job ID

parts	getRequestResponse element model:RequestMessagePart
used by	Operation getRequest in PortType EddsPortType
source	<wsdl:message name="getRequestResponse"> <wsdl:documentation>The message contains the original request associated to the given

	job ID</wsdl:documentation> <wsdl:part name="getRequestResponse" element="model:RequestMessagePart"/> </wsdl:message>
--	---

9.3.51 message getResponse

The message contains the details of a "response" request (a job ID)

parts	getResponse element model:DownloadList
used by	Operation getResponse in PortType EddsPortType
source	<wsdl:message name="getResponse"> <wsdl:documentation> Get the response file(s) for a request. If the file list is empty, all response files are returned.</wsdl:documentation> <wsdl:part name="getResponse" element="model:DownloadList"/> </wsdl:message>

9.3.52 message getResponseResponse

The message contains the response file associated to a specific request

parts	getResponseResponse element model:ResponseData
used by	Operation getResponse in PortType EddsPortType
source	<wsdl:message name="getResponseResponse"> <wsdl:documentation>The message contains the response file associated to a specific request</wsdl:documentation> <wsdl:part name="getResponseResponse" element="model:ResponseData"/> </wsdl:message>

9.3.53 message getRoles

The message contains the name of the mission to get the roles for

parts	getRoles element model:MissionName
used by	Operation getRoles in PortType EddsPortType
source	<wsdl:message name="getRoles"> <wsdl:documentation>The message contains the name of the mission to get the roles for</wsdl:documentation> <wsdl:part name="getRoles" element="model:MissionName"/> </wsdl:message>

9.3.54 message getRolesResponse

The list of roles for the specified mission

parts	getRolesResponse
-------	-------------------------

	element model:RolesList
used by	Operation getRoles in PortType EddsPortType
source	<code><wsdl:message name="getRolesResponse"> <wsdl:documentation>The list of roles for the specified mission</wsdl:documentation> <wsdl:part name="getRolesResponse" element="model:RolesList"/> </wsdl:message></code>

9.3.55 message getStatus

The message contains the details of a "status" request (a job ID)

parts	getStatus element model:JobIdPart
used by	Operation getStatus in PortType EddsPortType
source	<code><wsdl:message name="getStatus"> <wsdl:documentation>The message contains the details of a "status" request (a job ID)</wsdl:documentation> <wsdl:part name="getStatus" element="model:JobIdPart"/> </wsdl:message></code>

9.3.56 message getStatuses

Gets the status of a list of jobs

parts	getStatus element model:JobListIdPart
used by	Operation getStatuses in PortType EddsPortType
source	<code><wsdl:message name="getStatuses"> <wsdl:documentation>Gets the status of a list of jobs</wsdl:documentation> <wsdl:part name="getStatus" element="model:JobListIdPart"/> </wsdl:message></code>

9.3.57 message getStatusesResponse

Returns the allocated unique ID for the request. This is used to take the messages off the topic. The ID is a random string, not a Request ID.

parts	getStatusesResponse element model:JobIdPart
used by	Operation getStatuses in PortType EddsPortType
source	<code><wsdl:message name="getStatusesResponse"> <wsdl:documentation>Returns the allocated unique ID for the request. This is used to take the messages off the topic. The ID is a random string, not a Request ID.</wsdl:documentation> <wsdl:part name="getStatusesResponse" element="model:JobIdPart"/> </wsdl:message></code>

9.3.58 message getStatusResponse

The message contains the acknowledgement of a specific request

parts	getStatusResponse element model:AcknowledgementPart
used by	Operation getStatus in PortType EddsPortType
source	<code><wsdl:message name="getStatusResponse"></code> <code><wsdl:documentation></code> The message contains the acknowledgement of a specific request <code></wsdl:documentation></code> <code><wsdl:part name="getStatusResponse" element="model:AcknowledgementPart"/></code> <code></wsdl:message></code>

9.3.59 message getTransformations

The message contains the mission name and request type

parts	getTransformations element model:TransformationsRequest
used by	Operation getTransformations in PortType EddsPortType
source	<code><wsdl:message name="getTransformations"></code> <code><wsdl:documentation></code> The message contains the mission name and request type <code></wsdl:documentation></code> <code><wsdl:part name="getTransformations" element="model:TransformationsRequest"/></code> <code></wsdl:message></code>

9.3.60 message getTransformationsResponse

The message contains a list of available transformations

parts	getTransformationsResponse element model:TransformationsList
used by	Operation getTransformations in PortType EddsPortType
source	<code><wsdl:message name="getTransformationsResponse"></code> <code><wsdl:documentation></code> The message contains a list of available transformations <code></wsdl:documentation></code> <code><wsdl:part name="getTransformationsResponse" element="model:TransformationsList"/></code> <code></wsdl:message></code>

9.3.61 message getUserAccountDetails

The message contains a username

parts	getUserAccountDetails element model:UserName
used by	Operation getUserAccountDetails in PortType EddsPortType
source	<code><wsdl:message name="getUserAccountDetails"></code> <code><wsdl:documentation></code> The message contains a username <code></wsdl:documentation></code> <code><wsdl:part name="getUserAccountDetails" element="model:UserName"/></code>

	<code></wsdl:message></code>
--	------------------------------------

9.3.62 message getUserAccountDetailsResponse

The message contains the user details associated to the username as defined in the LDAP

parts	getUserAccountDetailsResponse element model:UserAccountDetails
used by	Operation getUserAccountDetails in PortType EddsPortType
source	<code><wsdl:message name="getUserAccountDetailsResponse"></code> <code><wsdl:documentation></code> The message contains the user details associated to the username as defined in the LDAP <code></wsdl:documentation></code> <code><wsdl:part name="getUserAccountDetailsResponse"</code> <code>element="model:UserAccountDetails"/></code> <code></wsdl:message></code>

9.3.63 message getUserQuota

The message contains a mission's name

parts	getUserQuota element model:MissionName
used by	Operation getUserQuota in PortType EddsPortType
source	<code><wsdl:message name="getUserQuota"></code> <code><wsdl:documentation></code> The message contains a mission's name <code></wsdl:documentation></code> <code><wsdl:part name="getUserQuota" element="model:MissionName"/></code> <code></wsdl:message></code>

9.3.64 message getUserQuotaResponse

The message contains the user quota usage details

parts	getUserQuotaResponse element model:UserQuotaDetails
used by	Operation getUserQuota in PortType EddsPortType
source	<code><wsdl:message name="getUserQuotaResponse"></code> <code><wsdl:documentation></code> The message contains the user quota usage details <code></wsdl:documentation></code> <code><wsdl:part name="getUserQuotaResponse" element="model:UserQuotaDetails"/></code> <code></wsdl:message></code>

9.3.65 message getUsers

Flag to indicate whether to fetch all users or not

parts	getUsers element model:IncludeAll
-------	--

used by	Operation getUsers in PortType EddsPortType
source	<wsdl:message name="getUsers"> <wsdl:documentation>Flag to indicate whether to fetch all users or not</wsdl:documentation> <wsdl:part name="getUsers" element="model:IncludeAll"/> </wsdl:message>

9.3.66 message getUsersResponse

The list of users

parts	getUsersResponse element model:UsersList
used by	Operation getUsers in PortType EddsPortType
source	<wsdl:message name="getUsersResponse"> <wsdl:documentation>The list of users</wsdl:documentation> <wsdl:part name="getUsersResponse" element="model:UsersList"/> </wsdl:message>

9.3.67 message hasPermissionToSeeAllUsers

The message contains a username

parts	hasPermissionToSeeAllUsers element model:UserName
used by	Operation hasPermissionToSeeAllUsers in PortType EddsPortType
source	<wsdl:message name="hasPermissionToSeeAllUsers"> <wsdl:documentation>The message contains a username</wsdl:documentation> <wsdl:part name="hasPermissionToSeeAllUsers" element="model:UserName"/> </wsdl:message>

9.3.68 message login

The message contains the user credentials needed for the authentication

parts	credentials element model:UserCredentials
used by	Operation login in PortType EddsPortType
source	<wsdl:message name="login"> <wsdl:documentation>The message contains the user credentials needed for the authentication</wsdl:documentation> <wsdl:part name="credentials" element="model:UserCredentials"/> </wsdl:message>

9.3.69 message loginResponse

The message returns the user credentials enriched with the session ID

parts	credentials element model:UserCredentials
used by	Operation logIn in PortType EddsPortType
source	<code><wsdl:message name="logInResponse"> <wsdl:documentation>The message returns the user credentials enriched with the session ID</wsdl:documentation> <wsdl:part name="credentials" element="model:UserCredentials"/> </wsdl:message></code>

9.3.70 message logOut

Empty message

parts	
used by	Operation logOut in PortType EddsPortType
source	<code><wsdl:message name="logOut"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message></code>

9.3.71 message passwordExpiredFault

The message contains details related to a authentication fault

parts	passwordExpiredFault element model:PasswordExpiredFault
used by	Operation logIn in PortType EddsPortType
source	<code><wsdl:message name="passwordExpiredFault"> <wsdl:documentation>The message contains details related to a authentication fault</wsdl:documentation> <wsdl:part name="passwordExpiredFault" element="model:PasswordExpiredFault"/> </wsdl:message></code>

9.3.72 message permissionToSeeAllUsersResponse

True if session has expired (or not started)

parts	permissionToSeeAllUsersResponse element model:PermissionToSeeAllUsers
used by	Operation hasPermissionToSeeAllUsers in PortType EddsPortType
source	<code><wsdl:message name="permissionToSeeAllUsersResponse"> <wsdl:documentation>True if session has expired (or not started)</wsdl:documentation> <wsdl:part name="permissionToSeeAllUsersResponse" element="model:PermissionToSeeAllUsers"/> </wsdl:message></code>

9.3.73 message requestFault

The message contains details related to a request fault

parts	requestFault element model:RequestFault
used by	Operation batchRequest in PortType EddsPortType Operation streamRequest in PortType EddsPortType Operation getStatus in PortType EddsPortType Operation getStatuses in PortType EddsPortType Operation getResponse in PortType EddsPortType Operation getRequest in PortType EddsPortType Operation getStreamRequest in PortType EddsPortType Operation startStreamData in PortType EddsPortType Operation getParamDefinitions in PortType EddsPortType Operation getFarcCatalogue in PortType EddsPortType Operation cancel in PortType EddsPortType Operation suspend in PortType EddsPortType Operation resume in PortType EddsPortType Operation deleteData in PortType EddsPortType Operation deleteRequest in PortType EddsPortType Operation getJobs in PortType EddsPortType Operation getJobsWithStatus in PortType EddsPortType Operation getJobsWithStatusAndLimit in PortType EddsPortType Operation getHistoricalLog in PortType EddsPortType Operation stopHistoricalLogRetrieval in PortType EddsPortType Operation getTransformations in PortType EddsPortType Operation accountRequest in PortType EddsPortType Operation getUsers in PortType EddsPortType Operation getMissions in PortType EddsPortType Operation getMissionsAndDomains in PortType EddsPortType Operation getRoles in PortType EddsPortType Operation getUserAccountDetails in PortType EddsPortType Operation getMissionDetails in PortType EddsPortType Operation getUserQuota in PortType EddsPortType Operation hasPermissionToSeeAllUsers in PortType EddsPortType
source	<code><wsdl:message name="requestFault"> <wsdl:documentation>The message contains details related to a request fault</wsdl:documentation> <wsdl:part name="requestFault" element="model:RequestFault"/> </wsdl:message></code>

9.3.74 message reSubmit

The message contains the details of a request to re-submit(a job ID)

parts	reSubmit element model:JobIdPart
source	<code><wsdl:message name="reSubmit"> <wsdl:documentation>The message contains the details of a request to re-submit(a job ID)</wsdl:documentation> <wsdl:part name="reSubmit" element="model:JobIdPart"/> </wsdl:message></code>

9.3.75 message reSubmitResponse

The message contains job IDs associated to a batch request

parts	reSubmitResponse element model:JobListIdPart
source	<code><wsdl:message name="reSubmitResponse"> <wsdl:documentation>The message contains job IDs associated to a batch request</wsdl:documentation> <wsdl:part name="reSubmitResponse" element="model:JobListIdPart"/> </wsdl:message></code>

9.3.76 message sessionFault

The message contains details related to a session fault

parts	sessionFault element model:SessionFault
used by	Operation batchRequest in PortType EddsPortType Operation streamRequest in PortType EddsPortType Operation getStatus in PortType EddsPortType Operation getStatuses in PortType EddsPortType Operation getResponse in PortType EddsPortType Operation getRequest in PortType EddsPortType Operation getStreamRequest in PortType EddsPortType Operation startStreamData in PortType EddsPortType Operation stopStreamData in PortType EddsPortType Operation getParamDefinitions in PortType EddsPortType Operation getFarcCatalogue in PortType EddsPortType Operation cancel in PortType EddsPortType Operation suspend in PortType EddsPortType Operation resume in PortType EddsPortType Operation deleteData in PortType EddsPortType Operation deleteRequest in PortType EddsPortType Operation getJobs in PortType EddsPortType Operation getJobsWithStatus in PortType EddsPortType Operation getJobsWithStatusAndLimit in PortType EddsPortType Operation getHistoricalLog in PortType EddsPortType Operation stopHistoricalLogRetrieval in PortType EddsPortType Operation getTransformations in PortType EddsPortType Operation checkUserPassword in PortType EddsPortType Operation accountRequest in PortType EddsPortType Operation getUsers in PortType EddsPortType Operation getMissions in PortType EddsPortType Operation getMissionsAndDomains in PortType EddsPortType Operation getRoles in PortType EddsPortType Operation getUserAccountDetails in PortType EddsPortType Operation getMissionDetails in PortType EddsPortType Operation getUserQuota in PortType EddsPortType Operation hasPermissionToSeeAllUsers in PortType EddsPortType
source	<code><wsdl:message name="sessionFault"> <wsdl:documentation>The message contains details related to a session</code>

	<code>fault</wsdl:documentation></code> <code><wsdl:part name="sessionFault" element="model:SessionFault"/></code> <code></wsdl:message></code>
--	---

9.3.77 message streamRequest

The message contains the details of a stream request

parts	streamRequest element model:StreamRequestMessagePart
used by	Operation streamRequest in PortType EddsPortType
source	<code><wsdl:message name="streamRequest"></code> <code><wsdl:documentation></code> The message contains the details of a stream request <code></wsdl:documentation></code> <code><wsdl:part name="streamRequest" element="model:StreamRequestMessagePart"/></code> <code></wsdl:message></code>

9.3.78 message streamRequestResponse

The message contains job ID associated to a stream request

parts	streamRequestResponse element model:JobIdPart
used by	Operation streamRequest in PortType EddsPortType
source	<code><wsdl:message name="streamRequestResponse"></code> <code><wsdl:documentation></code> The message contains job ID associated to a stream request <code></wsdl:documentation></code> <code><wsdl:part name="streamRequestResponse" element="model:JobIdPart"/></code> <code></wsdl:message></code>

9.3.79 message startStreamData

Starts the incoming stream of the specified active stream request to the client.

parts	startStreamData element model:JobIdPart
used by	Operation startStreamData in PortType EddsPortType
source	<code><wsdl:message name="startStreamData"></code> <code><wsdl:documentation></code> Starts the incoming stream of the specified active stream request to the client. <code></wsdl:documentation></code> <code><wsdl:part name="startStreamData" element="model:JobIdPart"/></code> <code></wsdl:message></code>

9.3.80 message startStreamDataResponse

Empty message

parts	
used by	Operation startStreamData in PortType EddsPortType

source	<wsdl:message name="startStreamDataResponse"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message>
--------	---

9.3.81 message stopHistoricalLogRetrieval

Stops the retrieval of historical log messages. Requires the request ID provided by getHistoricalLog.

parts	stopHistoricalLogRetrieval element model:JobIdPart
used by	Operation stopHistoricalLogRetrieval in PortType EddsPortType
source	<wsdl:message name="stopHistoricalLogRetrieval"> <wsdl:documentation>Stops the retrieval of historical log messages. Requires the request ID provided by getHistoricalLog.</wsdl:documentation> <wsdl:part name="stopHistoricalLogRetrieval" element="model:JobIdPart"/> </wsdl:message>

9.3.82 message stopHistoricalLogRetrievalResponse

Empty message

parts	
used by	Operation stopHistoricalLogRetrievalResponse in PortType EddsPortType
source	<wsdl:message name="stopHistoricalLogRetrievalResponse"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message>

9.3.83 message stopStreamData

Stops the incoming stream of the specified active stream request to the client.

parts	stopStreamData element model:JobIdPart
used by	Operation stopStreamData in PortType EddsPortType
source	<wsdl:message name="stopStreamData"> <wsdl:documentation>Stops the incoming stream of the specified active stream request to the client.</wsdl:documentation> <wsdl:part name="stopStreamData" element="model:JobIdPart"/> </wsdl:message>

9.3.84 message getStreamRequest

The message contains the details of a "stream request" request (a job ID)

parts	getStreamRequest element model:JobIdPart
used by	Operation getStreamRequest in PortType EddsPortType
source	<wsdl:message name="getStreamRequest"> <wsdl:documentation>The message contains the details of a "stream request" request (a job ID)</wsdl:documentation> <wsdl:part name="getStreamRequest" element="model:JobIdPart"/> </wsdl:message>

9.3.85 message getStreamRequestResponse

The message contains the original stream request associated to the given job ID

parts	getStreamRequestResponse element model:StreamRequestMessagePart
used by	Operation getStreamRequest in PortType EddsPortType
source	<code><wsdl:message name=" getStreamRequestResponse "></code> <code><wsdl:documentation></code> The message contains the original stream request associated to the given job ID <code></wsdl:documentation></code> <code><wsdl:part name=" getStreamRequestResponse " element="model:StreamRequestMessagePart"/></code> <code></wsdl:message></code>

9.3.86 message getAlias

Get aliases for the mission

parts	getAlias element model:AliasRequest
used by	Operation getAlias in PortType EddsPortType
source	<code><wsdl:message name=" getAlias"></code> <code><wsdl:documentation></code> Get aliases for the mission <code></wsdl:documentation></code> <code><wsdl:part name=" getAlias" element="model:AliasRequest "/></code> <code></wsdl:message></code>

9.3.87 message getAliasResponse

Returns the list of aliases for the mission

parts	getAliasResponse element model:ListAliasInfo
used by	Operation getAlias in PortType EddsPortType
source	<code><wsdl:message name=" getAliasResponse"></code> <code><wsdl:documentation></code> Returns the list of aliases for the mission. <code></wsdl:documentation></code> <code><wsdl:part name=" aliasResponse" element="model:ListAliasInfo"/></code> <code></wsdl:message></code>

9.3.88 message checkSession

Empty message

parts	
used by	Operation hasSessionExpired in PortType EddsPortType
source	<code><wsdl:message name=" checkSession"></code> <code><wsdl:documentation></code> Empty message <code></wsdl:documentation></code> <code></wsdl:message></code>

9.3.89 message stopStreamDataResponse

Empty message

parts	
-------	--

used by	Operation stopStreamData in PortType EddsPortType
source	<pre><wsdl:message name="stopStreamDataResponse"> <wsdl:documentation>Empty message</wsdl:documentation> </wsdl:message></pre>

9.4 Bindings

The EDDS binding describes how the service is bound to the messaging protocol.

type	ws:EddsPortType
extensibility	<pre><soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/></pre>
operations	batchRequest <i>This operation perform a batch request on the EDDS erver. The request is asynchonomous and its output is the job ID or an error in case of failure</i> extensibility<pre><soap:operation soapAction="http://edds.egos.esa/batchrequest/batchRequest" style="document"/></pre> input<pre>name="batchRequest"<soap:body use="literal"/></pre> output<pre>name="batchRequestResponse"<soap:body use="literal"/></pre> streamRequest <i>This operation perform a stream request on the EDDS erver. The request is asynchonomous and its output is the job ID or an error in case of failure</i> extensibility<pre><soap:operation soapAction="http://edds.egos.esa/streamrequest/streamRequest" style="document"/></pre> input<pre>name="streamRequest"<soap:body use="literal"/></pre> output<pre>name="streamRequestResponse"<soap:body use="literal"/></pre> getStatus <i>This operation provides the status of a specific request. The operation takes as input a job ID and return an acknowledgement</i> extensibility<pre><soap:operation soapAction="http://edds.egos.esa/batchrequest/getStatus" style="document"/></pre> input<pre>name="getStatus"<soap:body use="literal"/></pre> output<pre>name="getStatusResponse"<soap:body use="literal"/></pre> reSubmit <i>This operation re-submits a request. The operation takes as input a job ID and its output is the list of job IDs</i> extensibility<pre><soap:operation soapAction="http://edds.egos.esa/batchrequest/reSubmit" style="document"/></pre> input<pre>name="reSubmit"<soap:body use="literal"/></pre> output<pre>name="reSubmitResponse"<soap:body use="literal"/></pre> getStatuses

	<i>Gets the status of a list of jobs. The operation takes as input a list of job IDs and returns a java.util.ArrayList of type RequestInfoRec encoded in Base64</i> extensi bility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getStatuses" style="document"/> input name="getStatuses"<soap:body use="literal"/> output name="getStatusesResponse"<soap:body use="literal"/> getResponse <i>This operation returns the result file of abatch request. The response is available if the delivery method of the associated batch request was "server delivery". The method is synchronous.</i> extensi bility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getResponse" style="document"/> input name="getResponse"<soap:body use="literal"/> output name="getResponseResponse"<soap:body use="literal"/> getRequest <i>This operation provides the template of an existing batch request. The operation takes in input the batch request job ID and returns as output the corresponding XML. The method is synchronous.</i> extensi bility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getRequest" style="document"/> input name="getRequest"<soap:body use="literal"/> output name="getRequestResponse"<soap:body use="literal"/> getStreamRequest <i>This operation provides the template of an existing batch request. The operation takes in input the batch request job ID and returns as output the corresponding XML. The method is synchronous.</i> extensi bility <soap:operation soapAction="http://edds.egos.esa/streamrequest/getStreamRequest" style="document"/> input name="getStreamRequest"<soap:body use="literal"/> output name="getStreamRequestResponse"<soap:body use="literal"/> startStreamData <i>Starts the CometD forwarder of the specified stream request</i> extensi bility <soap:operation soapAction="http://edds.egos.esa/streamrequest/startStreamData" style="document"/> input name="startStreamData"<soap:body use="literal"/> output name="startStreamDataResponse"<soap:body use="literal"/> stopStreamData <i>Stops the CometD forwarder of the specified stream request</i> extensi bility <soap:operation soapAction="http://edds.egos.esa/streamrequest/stopStreamData" style="document"/> input name="stopStreamData"<soap:body use="literal"/>
--	--

	output name="stopStreamDataResponse"<soap:body use="literal"/> getParamDefinitions <i>Gets the parameter definitions for the specified mission. The method is synchronous.</i> extensi <soap:operation bility soapAction="http://edds.egos.esa/batchrequest/getParamDefinitions" style="document"/> input name="getParamDefinitions"<soap:body use="literal"/> output name="getParamDefinitionsResponse"<soap:body use="literal"/> getCatalogue <i>Gets the catalogue for the specified data source. The method is asynchronous.</i> extensi <soap:operation bility soapAction="http://edds.egos.esa/batchrequest/getCatalogue" style="document"/> input name="getCatalogue"<soap:body use="literal"/> output name="getCatalogueResponse"<soap:body use="literal"/> getFarcCatalogue <i>Gets the FARC catalogue for the specified mission and domain. The method is asynchronous. @deprecated</i> extensi <soap:operation bility soapAction="http://edds.egos.esa/batchrequest/getFarcCatalogue" style="document"/> input name="getFarcCatalogue"<soap:body use="literal"/> output name="getFarcCatalogueResponse"<soap:body use="literal"/> getDataspaces <i>Gets the dataspaces for the specified mission and request type. The method is synchronous.</i> extensi <soap:operation bility soapAction="http://edds.egos.esa/batchrequest/getDataspaces" style="document"/> input name="getDataspaces"<soap:body use="literal"/> output name="getDataspacesResponse"<soap:body use="literal"/> getAlias <i>Gets alias for the specified mission. The method is synchronous.</i> extensi <soap:operation bility soapAction="http://edds.egos.esa/batchrequest/getAlias" style="document"/> input name="getAlias"<soap:body use="literal"/> output name="getAliasResponse"<soap:body use="literal"/> getLastConsolidation <i>Gets the last consolidation time for the specified mission, request type and dataspace name if given. The method is synchronous.</i> extensi <soap:operation bility soapAction="http://edds.egos.esa/batchrequest/getLastConsolidation"
--	--

	style="document"/> input name="getLastConsolidation"<soap:body use="literal"/> output name="getLastConsolidationResponse"<soap:body use="literal"/>
	getAllowedRequestTypes Gets the request types that the currently logged-in user has permission to issue. The method is synchronous.. extensibility<soap:operation soapAction="http://edds.egos.esa/batchrequest/getAllowedRequestTypes" style="document"/> input name="getAllowedRequestTypes"<soap:body use="literal"/> output name="getAllowedRequestTypesResponse"<soap:body use="literal"/>
	cancel This operation cancel a scheduled request or try to interrupt an already executing request. The operation takes in input the job ID of the operation to cancel and return the job ID of the cancel request. The operation is asynchronous. extensibility<soap:operation soapAction="http://edds.egos.esa/batchrequest/cancel" style="document"/> input name="cancel"<soap:body use="literal"/> output name="cancelResponse"<soap:body use="literal"/>
	suspend This operation suspends an active request. The operation takes in input the job ID of the operation to suspend and return the job ID of the suspend request. The operation is asynchronous. extensibility<soap:operation soapAction="http://edds.egos.esa/batchrequest/suspend" style="document"/> input name="suspend"<soap:body use="literal"/> output name="suspendResponse"<soap:body use="literal"/>
	resume This operation resumes a suspended or failed request. The operation takes in input the job ID of the operation to resume and return the job ID of the resumed request. The operation is asynchronous. extensibility<soap:operation soapAction="http://edds.egos.esa/batchrequest/resume" style="document"/> input name="resume"<soap:body use="literal"/> output name="resumeResponse"<soap:body use="literal"/>
	deleteData This operation deletes the data related to a request. The operation is asynchronous. extensibility<soap:operation soapAction="http://edds.egos.esa/batchrequest/deleteData" style="document"/> input name="deleteData"<soap:body use="literal"/> output name="deleteDataResponse"<soap:body use="literal"/>
	deleteRequest

	<i>This operation deletes a request. The operation is asynchronous.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/deleteRequest" style="document"/> input name="deleteRequest"<soap:body use="literal"/> output name="deleteRequestResponse"<soap:body use="literal"/> getJobs <i>This operation returns the list of jobs over a defined time span. The operation takes in input the start and end date (where null is considered as ALL) and returns the list of operation that a user has right to read. The operation is synchronous.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getJobs" style="document"/> input name="getJobs"<soap:body use="literal"/> output name="getJobsResponse"<soap:body use="literal"/> getJobsWithStatus <i>This operation returns the list of jobs with their status over a defined time span. The operation takes in input the start and end date (where null is considered as ALL) and returns the list of operation that a user has right to read. The operation is synchronous. The message contains a java.util.ArrayList of type RequestInfoRec encoded in Base64</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getJobsWithStatus" style="document"/> input name="getJobsWithStatus"<soap:body use="literal"/> output name="getJobsWithStatusResponse"<soap:body use="literal"/> getJobsWithStatusAndLimit <i>This operation returns the list of jobs with their status over a defined time span. The operation takes in input the start and end date (where null is considered as ALL) and returns the list of operation that a user has right to read. The number of results returned can be limited by specifying an upper limit. Ommitting the limit returns all matching rows. The operation is synchronous. The message contains a java.util.ArrayList of type RequestInfoRec encoded in Base64</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getJobsWithStatusAndLimit" style="document"/> input name="getJobsWithStatusAndLimit"<soap:body use="literal"/> output name="getJobsWithStatusAndLimitResponse"<soap:body use="literal"/> getHistoricalLog <i>The operation return the historical log messages over a defined time span. The operation takes as input the start and end date (where null is considered as ALL) and returns a request ID used to listen to the resulting stream. The operation is asynchronous.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getHistoricalLog" style="document"/> input name="getHistoricalLog"<soap:body use="literal"/> output name="getHistoricalLogResponse"<soap:body use="literal"/> stopHistoricalLogRetrieval
--	--

	<i>The operation stops the historical log message retrieval. The operation takes as input the request ID of the stream.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/stopHistoricalLogRetrieval" style="document"/> input name="stopHistoricalLogRetrieval"<soap:body use="literal"/> output name="stopHistoricalLogRetrievalResponse"<soap:body use="literal"/>
	getTransformations <i>The operation returns the available transformations for the requet type.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/batchrequest/getTransformations" style="document"/> input name="getTransformations"<soap:body use="literal"/> output name="getTransformationsResponse"<soap:body use="literal"/>
	logIn <i>Login the user and start the session</i> extensibility <soap:operation soapAction="http://edds.egos.esa/authentication/logIn" style="document"/> input name="logIn"<soap:body use="literal"/> output name="logInResponse"<soap:body use="literal"/>
	hasSessionExpired <i>Checks if the user's session has expired, and return true if it has</i> extensibility <soap:operation soapAction="http://edds.egos.esa/authentication/hasSessionExpired" style="document"/> input name="checkSession"<soap:body use="literal"/> output name="checkSessionResponse"<soap:body use="literal"/>
	logOut <i>Logout the user and close the session</i> extensibility <soap:operation soapAction="http://edds.egos.esa/authentication/logOut" style="document"/> input name="logOut"<soap:body use="literal"/> output
	checkUserPassword <i>Check if the password supplied matches the user's current password. Used to double check the user's current password before allowing the user to change their password.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/authentication/checkUserPassword" style="document"/> input name="checkUserPassword"<soap:body use="literal"/> output name="checkUserPasswordResponse"<soap:body use="literal"/>
	accountRequest

	<i>This operation performs an account management request. The operation is asynchronous: the detail of the request are contained in the request itself and the result of the operation is a job id. In case of error the operation returns an error.</i> extensibility <soap:operation soapAction="http://edds.egos.esa/userrequest/accountRequest" style="document"/> input <soap:body use="literal"/> output <soap:body use="literal"/>
	getUsers <i>The operation return the list of users. The operation is synchronous. An error on the request will throw a fault</i> extensibility <soap:operation soapAction="http://edds.egos.esa/userrequest/getUsers" style="document"/> input name="getUsers"<soap:body use="literal"/> output name="getUsersResponse"<soap:body use="literal"/>
	getRoles <i>The operation return the list of roles the user has permission to use for the specified mission. The operation is synchronous. An error on the request will throw a fault</i> extensibility <soap:operation soapAction="http://edds.egos.esa/userrequest/getRoles" style="document"/> input name="getRoles"<soap:body use="literal"/> output name="getRolesResponse"<soap:body use="literal"/>
	getMissions <i>The operation return the list of missions. The operation is synchronous. An error on the request will throw a fault</i> extensibility <soap:operation soapAction="http://edds.egos.esa/userrequest/getMissions" style="document"/> input name="getMissions"<soap:body use="literal"/> output name="getMissionsResponse"<soap:body use="literal"/>
	getMissionsAndDomains <i>The operation return the list of missions with their domains. The operation is synchronous. An error on the request will throw a fault</i> extensibility <soap:operation soapAction="http://edds.egos.esa/userrequest/getMissionsAndDomains" style="document"/> input name="getMissionsAndDomains"<soap:body use="literal"/> output name="getMissionsAndDomainsResponse"<soap:body use="literal"/>
	getUserAccountDetails <i>The operation return the details associated to a specific user as defined in the LDAP. The operation is synchronous. An error on the request will throw a fault</i> extensibility <soap:operation soapAction="http://edds.egos.esa/userrequest/getUserAccountDetails"

	style="document"/> input name="getUserAccountDetails"<soap:body use="literal"/> output name="getUserAccountDetailsResponse"<soap:body use="literal"/> getMissionDetails The operation return the details associated to a specific mission as defined in the LDAP. The operation is synchronous. An error on the request will throw a fault extensi<soap:operation bility soapAction="http://edds.egos.esa/userrequest/getMissionDetails" style="document"/> input name="getMissionDetails"<soap:body use="literal"/> output name="getMissionDetailsResponse"<soap:body use="literal"/> getUserQuota Returns the details for the specified mission extensi<soap:operation bility soapAction="http://edds.egos.esa/userrequest/getUserQuota" style="document"/> input name="getUserQuota"<wsdl:documentation>The name of the mission</wsdl:documentation><soap:body use="literal"/> output name="getUserQuotaResponse"<wsdl:documentation>The quota details for the mission</wsdl:documentation><soap:body use="literal"/> getQuotaDetails Returns the quota for the specified user and mission extensi<soap:operation soapAction=" bility http://edds.egos.esa/userrequest/getQuotaDetails" style="document"/> input name="getQuotaDetails"<wsdl:documentation> The name of the user and mission</wsdl:documentation><soap:body use="literal"/> output name="getQuotaDetailsResponse"<wsdl:documentation> The quota details for the user and mission</wsdl:documentation><soap:body use="literal"/> hasPermissionToSeeAllUsers Checks if user has permission to see other users data extensi<soap:operation bility soapAction="http://edds.egos.esa/userrequest/hasPermissionToSeeAllUsers" style="document"/> input name="hasPermissionToSeeAllUsers"<wsdl:documentation>Username</wsdl:documentation><soap:body use="literal"/> output name="permissionToSeeAllUsersResponse"<wsdl:documentation>The result whether user has permission to see other users data or not</wsdl:documentation><soap:body use="literal"/>
used by	Port EddsPort in Service EddsService

9.5 EDDS REST API endpoints

An endpoint is a point at which an API -- the code that allows two software programs to communicate with each other -- connects with the software program. APIs work by sending requests for information from a web application or web server and receiving a response.

endpoint	localhost:8080/edds/EddsRestService/v1/
operations	batchRequest <i>This operation perform a batch request on the EDDS erver. The request is asynchonous and its output is the job ID or an error in case of failure</i> type POST input model: RequestMessagePart output model: JobListIdPart getStatus <i>This operation provides the status of a specific request. The operation takes as input a job ID and return an acknowledgement</i> type POST input model: JobIdPart output model: AcknowledgementPart streamRequest <i>This operation provides the status of a specific request. The operation takes as input a job ID and return an acknowledgement</i> type POST input model: StreamRequestMessagePart output model: JobIdPart reSubmit <i>This operation re-submits a request. The operation takes as input a job ID and its output is the list of job IDs</i> type POST input model: JobIdPart output model: AcknowledgementPart getStatuses <i>Gets the status of a list of jobs. The operation takes as input a list of job IDs and returns a java.util.ArrayList of type RequestInfoRec encoded in Base64</i> type POST input model: DownloadList output model: ResponseData getResponse <i>This operation returns the result file of abatch request. The response is available if the delivery method</i>

of the associated batch request was "server delivery". The method is synchronous.

type **POST**

input **model:** JobListIdPart

```
output  model: JobIdPart
```

getRequest

This operation provides the template of an existing batch request. The operation takes in input the batch request job ID and returns as output the corresponding XML. The method is synchronous.

type **POST**

```
input      model: JobIdPart
```

```
output  model: RequestMessagePart
```

getStreamRequest

This operation provides the template of an existing batch request. The operation takes in input the batch request job ID and returns as output the corresponding XML. The method is synchronous.

type **POST**

input **model:** JobListIdPart

```
output  model: StreamRequestMessagePart
```

startStreamData

Starts the CometD forwarder of the specified stream request

type **POST**

input **model:** JobIdPart

```
output {"status": "Stream data successfully started."}
```

stopStreamData

Stops the CometD forwarder of the specified stream request

type **POST**

```
input  model: JobIdPart
```

```
output {"status": "Stream data successfully stopped."}
```

getParamDefinitions

Gets the parameter definitions for the specified mission. The method is synchronous.

type **POST**

```
input  {"mission": "<String>"}
```

```
output  model: JobIdPart
```

	getCatalogue <i>Gets the catalogue for the specified data source. The method is asynchronous.</i> type POST input model: CatalogueRequest output model: JobIdPart getFarcCatalogue <i>Gets the FARC catalogue for the specified mission and domain. The method is asynchronous.</i> <i>@deprecated</i> type POST input model: FarcCatalogueRequest output model: JobIdPart getDataspaces <i>Gets the dataspaces for the specified mission and request type. The method is synchronous.</i> type POST input model: FarcCatalogueRequest output model: JobIdPart getAlias <i>Gets alias for the specified mission. The method is synchronous.</i> type POST input model: AliasRequest output model: ListAliasInfo getLastConsolidation <i>Gets the last consolidation time for the specified mission, request type and dataspace name if given. The method is synchronous.</i> type POST input model: LastConsolidationRequest output {"status": "<long>"} getAllowedRequestTypes <i>Gets the request types that the currently logged-in user has permission to issue. The method is synchronous..</i> type GET output model: AllowedRequestTypes cancel
--	--

	<i>This operation cancel a scheduled request or try to interrupt an already executing request. The operation takes in input the job ID of the operation to cancel and return the job ID of the cancel request. The operation is asynchronous.</i> type POST input model: CancelPartList output model: AcknowledgementPartList suspend <i>This operation suspends an active request. The operation takes in input the job ID of the operation to suspend and return the job ID of the suspend request. The operation is asynchronous.</i> type POST input model: SuspendPartList output model: AcknowledgementPartList resume <i>This operation resumes a suspended or failed request. The operation takes in input the job ID of the operation to resume and return the job ID of the resumed request. The operation is asynchronous.</i> type POST input model: ResumePartList output model: AcknowledgementPartList deleteData <i>This operation deletes the data related to a request. The operation is asynchronous.</i> type POST input model: JobIdPartList output model: AcknowledgementPartList deleteRequest <i>This operation deletes a request. The operation is asynchronous.</i> type POST input model: JobIdPartList output model: AcknowledgementPartList getJobs <i>This operation returns the list of jobs over a defined time span. The operation takes in input the start and end date (where null is considered as ALL) and returns the list of operation that a user has right to read. The operation is synchronous.</i> type POST input model: TimeWindow output model: JobListIdPart
--	--

	getJobsWithStatus This operation returns the list of jobs with their status over a defined time span. The operation takes in input the start and end date (where null is considered as ALL) and returns the list of operation that a user has right to read. The operation is synchronous. The message contains a java.util.ArrayList of type RequestInfoRec encoded in Base64 type POST input model: TimeWindow output model: JobIdPart getJobsWithStatusAndLimit This operation returns the list of jobs with their status over a defined time span. The operation takes in input the start and end date (where null is considered as ALL) and returns the list of operation that a user has right to read. The number of results returned can be limited by specifying an upper limit. Ommitting the limit returns all matching rows. The operation is synchronous. The message contains a java.util.ArrayList of type RequestInfoRec encoded in Base64 type POST input model: JobsFilter output model: JobIdPart getHistoricalLog The operation return the historical log messages over a defined time span. The operation takes as input the start and end date (where null is considered as ALL) and returns a request ID used to listen to the resulting stream. The operation is asynchronous. type POST input model: TimeWindow output model: JobIdPart stopHistoricalLogRetrieval The operation stops the historical log message retrieval. The operation takes as input the request ID of the stream. type POST input model: JobIdPart output {“status”: “Service successfully stopped.”} getTransformations The operation returns the available transformations for the requet type. type POST input model: TransformationsRequest output model: TransformationsList logIn Login the user and start the session type POST
--	---

	input model: UserCredentials output {“token”: “<String >”}
	hasSessionExpired <i>Checks if the user's session has expired, and return true if it has</i> type GET output {“status”: “<Boolean>”}
	logOut <i>Logout the user and close the session</i> type GET output {“status”: “Successfully logged out.”}
	checkUserPassword <i>Check if the password supplied matches the user's current password. Used to double check the user's current password before allowing the user to change their password.</i> type POST input {“password”: “<String>”} output {“status”: “Password successfully checked.”}
	accountRequest <i>This operation performs an accaunt management request. The operation is asynchronous: the detail of the request are contained in the request itself and the result of the operation is a job id. In case of error the operation returns an error.</i> type POST input model: AccountRequestMessagePart output {“status”: “Account request successfully executed.”}
	getUsers <i>The operation return the list of users. The operation is synchronous. An error on the request will throw a fault</i> type POST input {“includeAll”: “<Boolean>”} output model: UsersList
	getRoles <i>The operation return the list of roles the user has permission to use for the specified mission. The operation is synchronous. An error on the request will throw a fault</i> type POST input {“missionName”: “<String>”}

	output model: RolesList getMissions <i>The operation return the list of missions. The operation is synchronous. An error on the request will throw a fault</i> type GET output model: MissionsList getMissionsAndDomains <i>The operation return the list of missions with their domains. The operation is synchronous. An error on the request will throw a fault</i> type GET output model: MissionsDomainList getUserAccountDetails <i>The operation return the details associated to a specific user as defined in the LDAP. The operation is synchronous. An error on the request will throw a fault</i> type POST input {“username”: “<String>”} output model: UserAccountDetails getMissionDetails <i>The operation return the details associated to a specific mission as defined in the LDAP. The operation is synchronous. An error on the request will throw a fault</i> type POST input {“missionName”: “<String>”} output model: MissionDetailsList getUserQuota <i>Returns the details for the specified mission</i> type POST input {“missionName”: “<String>”} output model: UserQuotaDetailsList getQuotaDetails <i>Returns the quota for the specified user and mission</i> type POST input model: UserMission output model: UserQuotaDetailsList
--	--

	hasPermissionToSeeAllUsers Checks if user has permission to see other users data type POST input {“username”: “<String>”} output {“status”: “<Boolean>”}
--	---

9.5.1 REST API Authentication

On top of the already existing authentication system based on session (unsecure), the REST API implementation provides a new authentication layer based on JWT.

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.

The current implementation signes the JWT tokens using HMAC algorithm.

In authentication, when the user successfully logs in using their credentials, a JSON Web Token will be returned.

Whenever the user wants to access a protected route or resource, the user agent should send the JWT, typically in the Authorization header using the Bearer schema. The content of the header should look like the following:

Authorization: Bearer <token>

The server's protected routes will check for a valid JWT in the Authorization header, and if it's present, the user will be allowed to access protected resources.

The key used for the encryption can be found into edds-deploy.properties: *edds.rest.api.jwt.encryption.key*.

Is suggested to change this key for each deployment.

9.5.2 REST API Testing

Approach 1:

Swagger API documentation is available at EDDS Home Page. Click on "EDDS REST Documentation".

This page contains the link to navigate to EDDS Swagger end point documentation. The login action should be performed first to get the authentication token. To authorize, click on the Authorize button. The value for authorization will be "Bearer #token#". The authorization can be done for all the end points together or seperately for each end points. The token would expire after some time and new one needs to be generated and re authorized.

The Swagger UI feature "Try it out" will let you to execute the end points. The inputs can be edited from the "Edit Value". The response will be available on "Execute" of request. Please read the main description of end points. For some end points, the description will have an example about the input for the end points.

Approach 2:

Since the implemented REST APIs are not yet used by ws-client, this section gives an overview on how to manually test new Rest web services.

Side by side with that, unit tests are implemented for most of the methods.

There are several tools for API testing. **Postman** is one of them.

Simple API calls can be executed using **Postman**.

Below is the **Postman** workspace.

In order to be able to get access to all the endpoints mentioned in the previous section, the user should use a valid Json Web Token.

To generate a JWT, the user should execute a REST API call for the login method .

To do so, the user should :

1. Set the HTTP request to POST
2. Input the link to the login API in the requested URL field
3. Add the needed JSON to the body : as specified in the login REST API description, the input should be of type UserCredentials. An example of the input body is given in the following image.
4. Click Send
5. Check the Status.
6. Check the Response field

If the user credentials are correct, the Status should be 200 OK and the output of the login service is a JWT. This JWT will be used to execute most of the other REST API calls.

If not, the Status is “401 Unauthorized” and the output of the login service is of type AuthenticationFaultMapper.

Once the login is successful, the user should save the JWT.

The service “hasSessionExpired” will be used for testing. To do so, the user should :

1. Set the HTTP request to GET
2. Add the string “Bearer “ + JWT to the value of the Authorization header
3. Input the link for the specific REST API in the url field
4. Click send

If the JWT is valid, the Status should be 200 OK and the output will be

If not, the Status is 401 Unauthorized and the output is the string “unauthorized access”.

A second example will be, the “getUsers” REST API.

To execute the getUsers REST API call, the user should :

1. Set the HTTP request to POST
2. Input the link in the requested URL field
3. Add the string “Bearer “ + JWT to the value of the Authorization header
4. Add the JSON { “includeAll”: “true” } as an example to the body of the request
5. Check the Status and the response of the API Call

The output should be the list of users.

If the JWT is not valid, the Status is “401 Unauthorized” and the output is the string “unauthorized access”.

If the input is not correct and the JWT is valid, the Status will be “500 Internal Server Error” and the output is of type RequestFaultMapper.

10. E-mail Interface

10.1 Data Elements

The EDDS allows users to request the delivery of acknowledgement messages by email. This implies that the receiving client system has access to a mail server.

10.2 Message Description

The email contains a configurable message that can be tailored by the mission. The acknowledgement message is in attachment to the email in XML format.

10.3 Communication Protocol

SMTP or SMTPs are used by the EDDS.

10.4 Errors

A SMTP/SMTPs client cannot verify the correct reception of an email. It is assumed that the configuration for the email has been properly set by the mission and that the email address given by the user is valid.

11. File Server Interface

11.1 Data Elements

The EDDS allows users to request the delivery of batch responses to one or more client hosted file server(s). On completion of the batch request, the EDDS will try to deliver the response to the addresses given in the corresponding request.

11.2 Message Description

A description of a File server delivery request can be found in section Appendix C

A description of a batch response can be found in section Appendix C

11.3 Communication Protocol

The EDDS uses the FTP protocol (or SFTP depending on the mission configuration) to deliver the batch response using the PUT method. The client's file server must support the PUT action.

11.4 Errors

If a batch response cannot be delivered to one or more of the remote FTP server(s) (after a number of retries) it may be deleted or retained (for a limited period) on the EDDS server as described in the original batch request.

12. Client API

The section describes the Java APIs provided by the EDDS application. The purpose of the APIs are to allow a client application access to the EDDS services. After the “package” Maven target has been run within the EDDS source directories, the JAR file can be found within the “edds-ws-client/target” folder. After the “install” target has been run, the library is copied to the local Maven repository. See the Configuration and Installation Guide (CIG) [RD-15] for information on how to build and deploy EDDS.

The EDDS APIs are derived from the definition of the EDDS Web Services. The API's implement a façade pattern on top of the classes generated from the WSDL files, which describe the EDDS services. They hide the implementation details of the services, like the connection to the web server.

EDDS Client API offers the following types of methods:

- EDDS Web Services client
- EDDS Push notifications client
- Utility methods

12.1 Edds Web Services Client API

The tables below describe the Client API. The class EddsService is a façade to the application. The methods available in the API match exactly those listed in Section 9.4.

12.1.1 Instantiating

Operation	Returns	Description
EddsService(String serverEndPoint)	EddsService	Creates an instance
setEndPointAddress(String address)	void	Sets the end-point to the address specified
connectToServer()	void	Initialises the service or reconnects to the server on a previously set address
setNotificationManager(NotificationManager notificationManager)	void	Sets an instance of the notification manager to be used with this EddsService

Table 30 - Instantiating

12.1.2 Session Management

Operation	Returns	Description
login(String userName, String password)	String	Creates session with server, also reconnects NotificationManager. Returns session ID.
hasSessionExpired()	boolean	Checks if the user's session has expired, or has not been created
checkUserPassword(String password)	void	Check if the password supplied matches the user's current password.
logout()	void	Log out the user and close the session.

Table 31 - Session Management

12.1.3 Issuing Requests

Operation	Returns	Description
batchRequest(RequestMessagePart parameters)	JobListIdPart	Submit a batch request.
streamRequest(StreamRequestMessagePart parameters)	JobIdPart	Perform a stream request.

Table 32 - Issuing Requests

12.1.4 Managing Requests

Operation	Returns	Description
cancel(CancelPartList cancelJobIds)	AcknowledgementPartList	Sends a Cancel request
reSubmit(JobIdPart job)	JobListIdPart	Resubmits the given request as a new request.
suspend(SuspendPartList suspendJobIds)	AcknowledgementPartList	Suspend request(s)
resume(ResumePartList resumeJobIds)	AcknowledgementPartList	Resume request(s)
deleteData(JobIdPartList requestIds)	AcknowledgementPartList	Delete response data
deleteRequest(JobIdPartList requestIds)	AcknowledgementPartList	Delete request(s) and their response data

Operation	Returns	Description
startStreamData(JobIdPart jobIdPart)	void	Start the stream from the web server to the client using CometD for the specified active stream request
stopStreamData(JobIdPart jobIdPart)	void	Stop the stream from the web server to the client using CometD for the specified active stream request

Table 33 - Managing Requests

12.1.5 Special Requests

Operation	Returns	Description
*getParamDefinitions(String mission)	JobIdPart	Initiates the retrieval of parameter definitions for the specified mission.
getDataspaces(String mission, String domain, DataSource dataSource)	List<String>	Gets the available dataspace from the EDDS server for the specified mission, domain (optional) and data source.
*getFarcCatalogue(String mission, String domain, String folder)	JobIdPart	Initiates the retrieval of a FARC Catalogue for the specified mission, domain and folder.
*getFsCatalogue(String mission, String folder)	String - the request ID	Initiates the retrieval of a File System Catalogue for the specified mission and folder
*getHistoricalLog(Date from, Date to)	JobIdPart	Return a list of historical log messages between the from and to date.
stopHistoricalLogRetrieval(JobIdPart requestId)	void	Stops the retrieval of historical log messages identified by the requestId.
getLastConsolidation(String mission, DataSource dataSource, String dataspace)	long	Gets the last consolidation time for the specified mission, data source and dataspace.
getAlias(String mission)	List<AliasType>	Get aliases definition for the specified mission.

Table 34 - Special Requests

* The response to this request is delivered through asynchronous notification to the push notifications client. See section 12.2.

12.1.6 Utility Methods for Requests

Operation	Returns	Description
getResponse(DownloadList downloadList)	ResponseData	Gets the response file(s) from EDDS File server.
getJobs(XMLGregorianCalendar start, XMLGregorianCalendar end)	JobListIdPart	Gets the jobs in the given time windows.
getJobsWithStatus(XMLGregorianCalendar start, XMLGregorianCalendar end)	JobIdPart	Gets the status of jobs in the given time windows.
getJobsWithStatusAndLimit(JobsFilter filter)	JobIdPart	Gets the status of jobs in the given time windows, limiting the number of rows returned by the limit specified.
getStatus(JobIdPart job)	AcknowledgementPart	Gets the acknowledgement.
getStatuses(JobListIdPart jobs)	JobIdPart	Gets the statuses of multiple jobs.
getRequest(JobIdPart job)	RequestMessagePart	Return the specified RequestMessagePart
getRequest(String job)	RequestMessagePart	Return the specified RequestMessagePart

Operation	Returns	Description
getStreamRequest(JobIdPart job)	StreamRequestMessagePart	Return the specified StreamRequestMessagePart

Table 35 - Utility Methods for Requests

12.1.7 Utility Methods for User Management

Operation	Returns	Description
getAllowedRequestTypes()	List <DataSourceAccess>	Gets the list of allowed request types with data sources that the user can submit
getMissions()	MissionsList	Gets all the mission names.
getMissionsAndDomains()	MissionsDomainList	Gets all the mission names and domains.
getMissionDetails(String mission)	MissionDetailsList	Gets all the details for the specified mission.
getRoles(String missionName)	RolesList	Gets all the roles for the specified mission
getUsers(boolean includeAll)	UsersList	Returns a the list of EDDS users
getUserAccountDetails(String username)	UserAccountDetails	Returns the full account details of the specified user
getUserQuota(String mission)	UserQuotaDetailsList	Gets the current user quota usage for the specified mission.
hasPermissionToSeeAllUsers(String username)	boolean	Checks if user has permission to see other users data.
getQuotaDetails(String user, String mission)	UserQuotaDetailsList	Gets the quota usage for the specified user and mission.
getTransformations(String mission, DataAccessDataElement type)	List<String>	Retrieves the list of available transformations for given mission and request type
accountRequest(AccountRequestMessagePart parameters)	void	Performs the specified changes in LDAP according to the contents of the AccountRequestMessagePart

Table 36 - Utility Methods for User Management

12.1.8 Edds Client API – code examples

This section provides some snippets of code in order to facilitate the developer in the usage of the library. The code is quite minimalistic since all the complexity of the SOAP protocol is hidden by the library. More examples can be found in the source code under edds-ws-client/src/test/java under esa.egos.edds.ws.client.examples.

Note that if you use the NotificationManager component in your client, then calling the **NotificationManager::disconnect()** method is mandatory, once you no longer need it. If you use EddsService then the call to the method **EddsService::logOut()** will disconnect the NotificationManager for you, so this call also becomes mandatory. If they are not called, then you risk keeping alive some threads dealing with connection that could prevent the program from closing or causing leaks.

12.1.8.1 Login

```
/**
 * Perform the login of a user and open a session on the web server.
 *
 * @param username The username
 * @param password The password
 * @throws WebServiceException
 * @throws MalformedURLException
 * @throws PasswordExpiredFault
```

```
    */
    public void logIn(String username, String password) throws
MalformedURLException, WebServiceException, PasswordExpiredFault
    {
        String serverAddress =
"http://10.48.29.159:8080/edds/EddsService?wsdl";
        eddsService = new EddsService(serverAddress);
        // These two lines are necessary only if notifications are used
        notificationManager = new NotificationManager(eddsService,
serverAddress);
        eddsService.setNotificationManager(notificationManager);
        // Initialize the connection to the web service
        eddsService.connectToServer();

        try
        {
            eddsService.logIn(username, password);
        }
        catch (AuthenticationFault ex)
        {
            // exception management
        }
    }
}
```

12.1.8.2 LogOut

```
/**
 * Close the session of the user
 *
 * @param username The username
 * @param password The password
 * @throws WebServiceException
 * @throws MalformedURLException
 */
public void logOut(String username, String password) throws
MalformedURLException, WebServiceException
{
    // we assume the login has been already performed
    eddsService.logOut();
}
```

12.1.8.3 GetStatus

```
/**
 * Return the status of a request
 *
 * @param jobIdPart The ID of the request
 * @throws WebServiceException
 * @throws MalformedURLException
 */
public void getStatus(JobIdPart jobIdPart) throws
MalformedURLException, WebServiceException
{
    try
    {
        // we assume the login has been already performed

        // we have as return the complete acknowledgement of the
request
        AcknowledgementPart acknowledgment =
eddsService.getStatus(jobIdPart);
    }
}
```



```

        // ...
    }
    catch (RequestFault ex)
    {
        // exception management
    }
    catch (SessionFault ex)
    {
        // exception management
    }
    catch (AuthorizationFault ex)
    {
        // exception management
    }
}

```

12.1.8.4 Get request statuses

```

/**
 * Return the request information of multiple requests
 *
 * @param jobListIdPart The IDs of the requests
 * @throws WebServiceException
 * @throws DatatypeConfigurationException
 * @throws ClassNotFoundException
 * @throws IOException
 * @throws TimeFormatException
 */
public void getJobsWithStatuses(JobListIdPart jobListIdPart) throws
WebServiceException, DatatypeConfigurationException, IOException,
    ClassNotFoundException, TimeFormatException
{
    try
    {
        // we assume the login has been already performed

        // Create a start time
        Date startTime = new Date(System.currentTimeMillis());
        GregorianCalendar startTimeGregorian = new
GregorianCalendar();
        startTimeGregorian.setTime(startTime);
        XMLGregorianCalendar startTimeXml =
DatatypeFactory.newInstance().newXMLGregorianCalendar(startTimeGregorian)
;

        // Create an end time
        Date endTime = new Date(System.currentTimeMillis());
        GregorianCalendar endTimeGregorian = new GregorianCalendar();
        endTimeGregorian.setTime(endTime);
        XMLGregorianCalendar endTimeXml =
DatatypeFactory.newInstance().newXMLGregorianCalendar(endTimeGregorian);

        // we have as return a job ID which can be used to identify
the response from CometD. The JobID is a random string, not a RequestId
        JobIdPart jobId = eddsService.getJobsWithStatus(startTimeXml,
endTimeXml);
    }
    catch (RequestFault ex)
    {

```

```

        // exception management
    }
    catch (SessionFault ex)
    {
        // exception management
    }
    catch (AuthorizationFault ex)
    {
        // exception management
    }
}

```

12.1.8.5 Perform BatchRequest

```

/**
 * Perform a batch request
 *
 * @param request The request message
 * @throws WebServiceException
 * @throws MalformedURLException
 */
public void batchRegeust(RequestMessagePart request) throws
MalformedURLException, WebServiceException
{
    try
    {
        // we assume the login has been already performed

        // the request is submitted and we receive the related ID
        JobListIdPart jobId = eddsService.batchRequest(request);

        // ...
    }
    catch (RequestFault ex)
    {
        // exception management
    }
    catch (SessionFault ex)
    {
        // exception management
    }
    catch (AuthorizationFault ex)
    {
        // exception management
    }
}

```

12.1.8.6 Perform StreamRequest

```

/**
 * Perform a stream request
 *
 * @param request The request message
 * @throws WebServiceException
 * @throws MalformedURLException
 */
public void streamRequest(StreamRequestMessagePart request) throws
MalformedURLException, WebServiceException
{
    try

```

```

    {
        // we assume the login has been already performed

        // the request is submitted and we receive the related ID
        JobListIdPart jobId = eddsService.streamRequest(request);

        // ...
    }
    catch (RequestFault ex)
    {
        // exception management
    }
    catch (SessionFault ex)
    {
        // exception management
    }
    catch (AuthorizationFault ex)
    {
        // exception management
    }
}

```

To receive the stream data, follow the example in Section 12.2.2, using `NotificationChannel.PARAM_STREAM` as the channel to listen to, and implement `NotificationListener <DarParamRec>`. Next call the method `startStreamData(requestId)` in the Client API to request that the Web Server starts to send the stream data to your application. When finished, ensure you call `stopStreamData(requestId)`. The `RequestId` to pass is the one returned to you from the above method - `streamRequest(request)`.

12.1.8.7 Cancel

```

/**
 * Cancels request(s)
 *
 * @param jobIds The request IDs
 * @throws WebServiceException
 * @throws MalformedURLException
 */
public void cancel(CancelPartList jobIds) throws
MalformedURLException, WebServiceException
{
    try
    {
        // we assume the login has been already performed

        // we have as return the complete acknowledgement of the
request(s)
        AcknowledgementPartList acknowledgements =
eddsService.cancel(jobIds);

        // ...
    }
    catch (RequestFault ex)
    {
        // exception management
    }
    catch (SessionFault ex)
    {
        // exception management
    }
    catch (AuthorizationFault ex)

```

```
    {  
        // exception management  
    }  
}
```

12.1.8.8 Suspend

```
/**  
 * Suspends a request(s)  
 *  
 * @param jobIds The request IDs  
 * @throws WebServiceException  
 * @throws MalformedURLExceptionException  
 */  
public void suspend(SuspendPartList jobIds) throws  
MalformedURLExceptionException, WebServiceException  
{  
    try  
    {  
        // we assume the login has been already performed  
  
        // we have as return the complete acknowledgement of the  
request(s) AcknowledgementPartList acknowledgements =  
eddsService.suspend(jobIds);  
  
        // ...  
    }  
    catch (RequestFault ex)  
    {  
        // exception management  
    }  
    catch (SessionFault ex)  
    {  
        // exception management  
    }  
    catch (AuthorizationFault ex)  
    {  
        // exception management  
    }  
}
```

12.1.8.9 Resume

```
/**  
 * Resumes a request(s)  
 *  
 * @param jobIds The request IDs  
 * @throws WebServiceException  
 * @throws MalformedURLExceptionException  
 */  
public void resume (ResumePartList jobIds) throws  
MalformedURLExceptionException, WebServiceException  
{  
    try  
    {  
        // we assume the login has been already performed  
  
        // we have as return the complete acknowledgement of the  
request(s)
```

```

        AcknowledgementPartList acknowledgements =
eddsService.resume(jobIds);

        // ...
    }
    catch (RequestFault ex)
    {
        // exception management
    }
    catch (SessionFault ex)
    {
        // exception management
    }
    catch (AuthorizationFault ex)
    {
        // exception management
    }
}

```

12.1.8.10 *Get parameter definitions*

```

public void getParamDefinitions(String mission)
{
    try
    {
        // we assume the login has been already performed

        JobIdPart jobId = eddsService.getParamDefinitions(mission);

        // ...
    }
    catch (RequestFault e)
    {
        // exception management
    }
    catch (SessionFault e)
    {
        // exception management
    }
    catch (AuthorizationFault e)
    {
        // exception management
    }
}

```

12.1.8.11 *Download response file*

```

/**
 * Save the response files for the given jobId locally
 *
 * @param jobId the request id
 */
public void saveResponseFiles(String jobId) throws
WebServiceException, RequestFault, SessionFault, AuthorizationFault,
FileNotFoundException, IOException
{
    DownloadList downloadList = new DownloadList();
    downloadList.setJob(jobId);
    List <ResponseDataElement> dataHandlers = eddsService.
getResponse(downloadList).getResponseDataElement();
}

```

```
// Save the response files
for (ResponseDataElement responseElem : dataHandlers)
{
    BufferedOutputStream bos = new BufferedOutputStream(new
FileOutputStream(new File(responseElem.getFileName()) ));
    responseElem.getResponse().writeTo(bos);
    bos.close();
}
}
```

12.1.9 EDDS Client API – setup a client

This section provides a thorough explanation for the setup of a Maven project containing a client for the EddsService façade. It is recommended to use *Eclipse IDE 2019-09, version 4.13.0* to setup the project.

The following EDDS dependencies are required for the project setup which are readily available on nexus for version 3.3.1:

- edds-common
- edds-ws-common
- edds-ws-client

12.1.9.1 Steps to be followed on Eclipse

This section provides details on the steps to be followed on Eclipse to setup the EDDS client. Open the Eclipse IDE and do the following:

- New Project Wizard with *File -> New -> Project*
- Select “*Maven Project*” as shown in Figure 3, and click “*Next*”
- Toggle the option “*Create a simple project (skip archetype selection)*” and choose a preferred project location, as shown in Figure 4
- Select a project location and click “*Next*”
- Provide a *groupId* and *artifactId* and *version* for your Maven project
- Also provide the *name* and *description*. The parent project can be left empty (as shown in Figure 5).

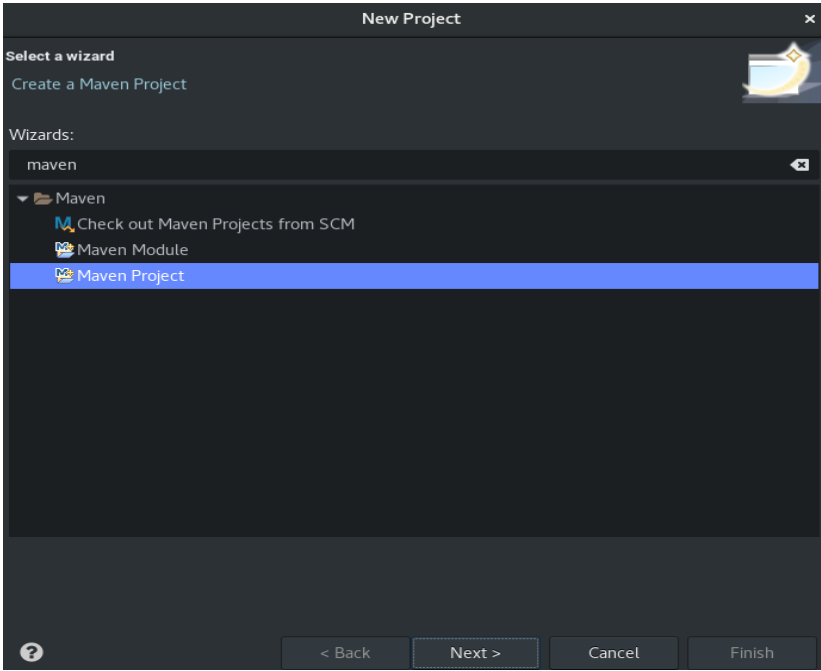


Figure 3 - Maven Project Setup

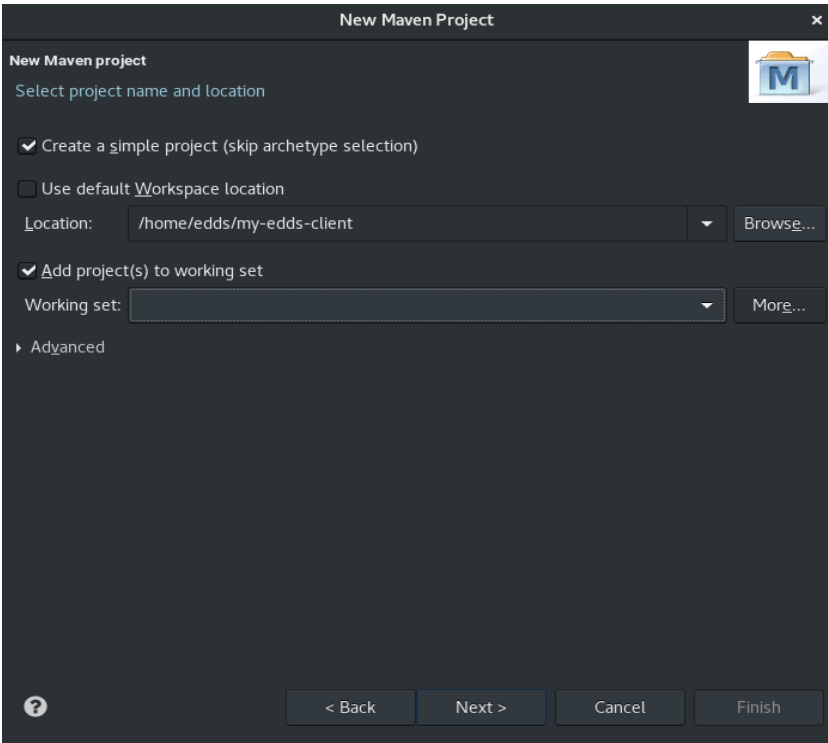


Figure 4 - Maven Project Setup Archetype

- Click “*Finish*” in this properties window.
- On the **Projects Explorer** view, setup *Project Natures* for the newly created project and by selecting “*Properties*” ->“*Project Natures*”
- If the *Java* and *Maven* natures are not present, add them

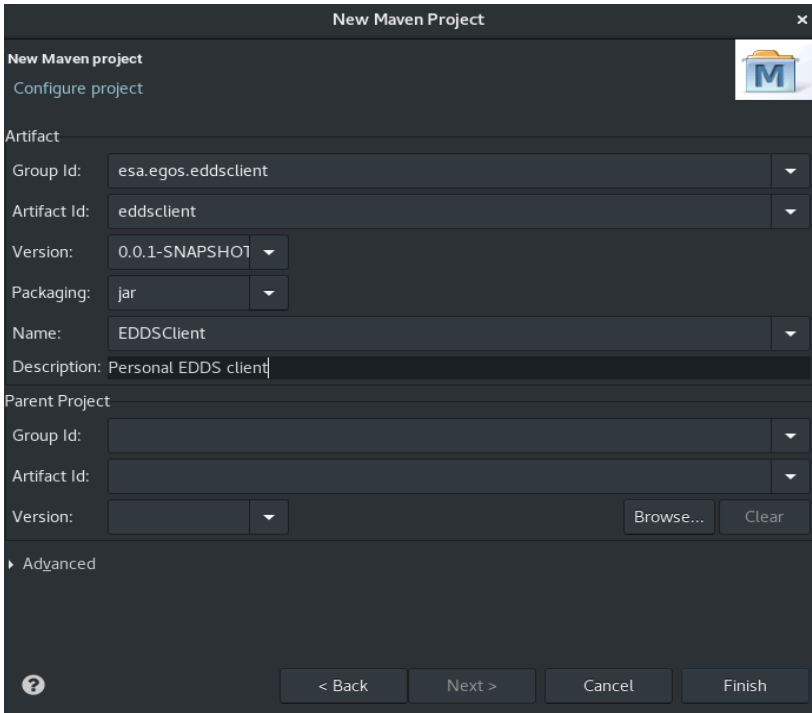


Figure 5 - Maven project setup: group and artifact id

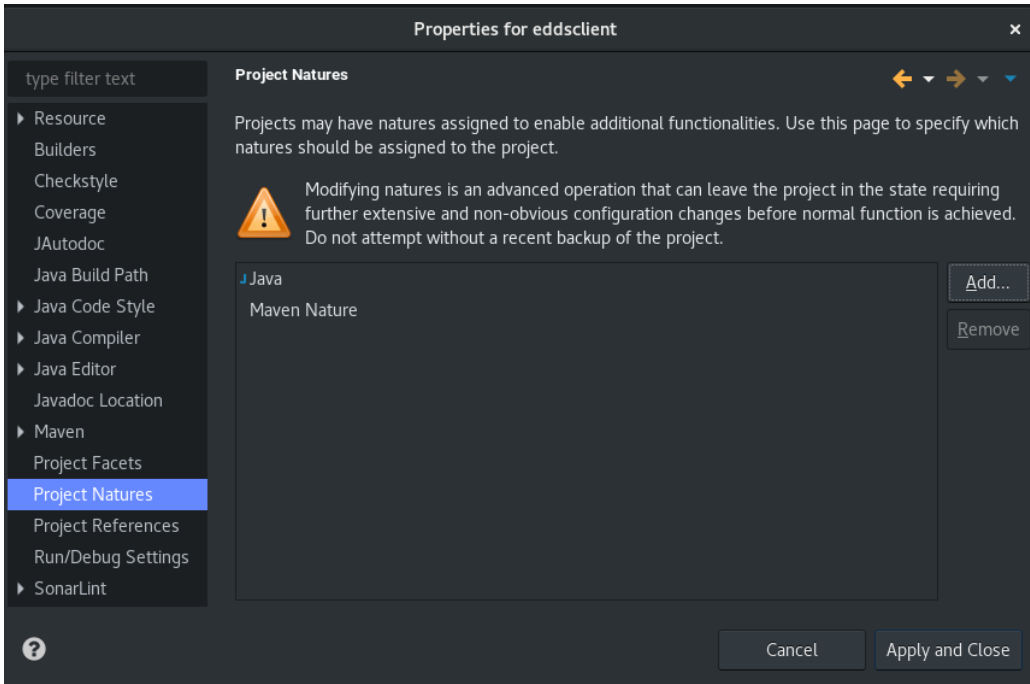


Figure 6 - Add Java Nature to a Maven Project

- The last configuration step is for the compiler compliance. EDDS is based on **Java 11** and also the client should maintain this compliance level. Choose “*Java Compiler*” and leave as flagged the option “*Enable Project specific settings*”. In the JDK Compliance, make sure the compiler compliance level is **11**, as shown in Figure 7 and press “*Apply*”.

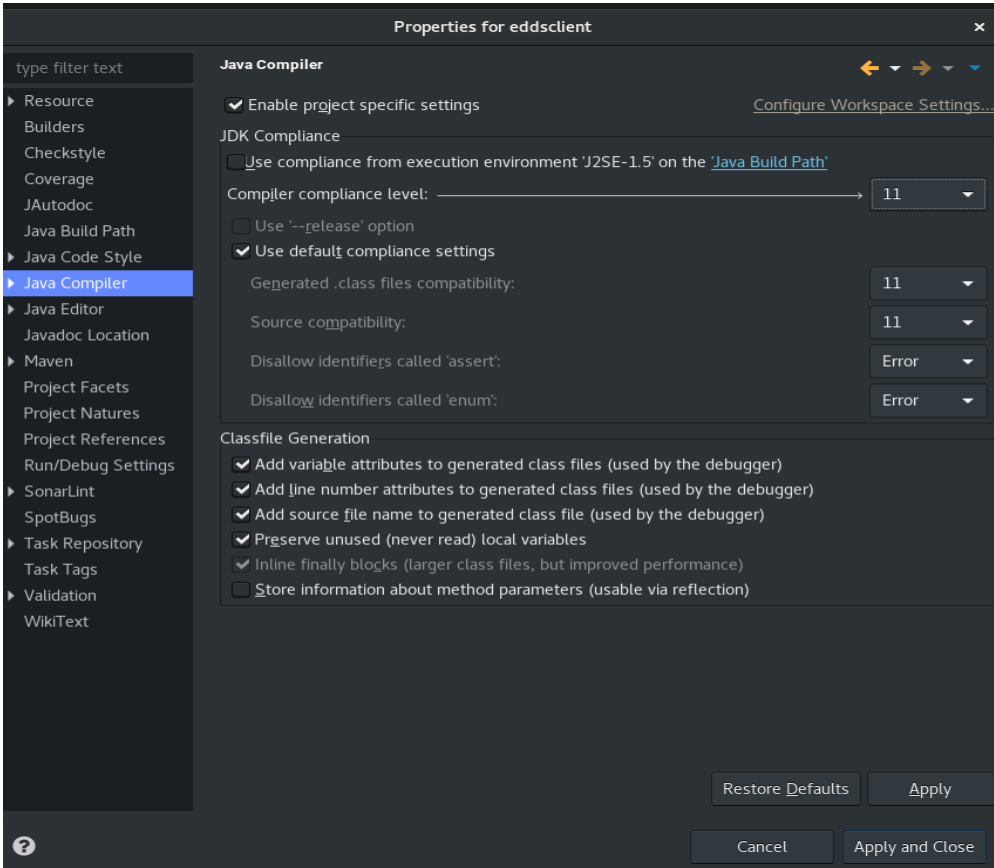


Figure 7 - Project Compiler Compliance Level

We will now configure the Maven dependencies needed for the project. These dependencies are summarised in Table 37.

Dependency name	Version
jaxws-api	2.3.0
edds-ws-client	3.6.0
edds-ws-client	3.6.0
edds-ws-common	3.6.0
edds-common	3.6.0
commons-codec	1.10
bayeux-api	3.1.5
jetty-client	9.4.40.v20210413
jetty-util-ajax	9.4.40.v20210413
cometd-java-client	9.3.25.v20180904
protobuf-java	3.7.0
maven-compiler-plugin	3.8.0

Table 37 - EDDS Client Maven Dependencies

Open the file pom.xml clicking on “Open With” -> “Maven POM Editor” and copy-paste the properties and dependencies from the following listing:

```
<properties>
  <edds.version>3.0.0</edds.version>
  <library-jetty.version>9.4.40.v20210413</library-jetty.version>
  <library-protobuf-java.version>3.7.0</library-protobuf-java.version>
  <library-commons-codec.version>1.10</library-commons-codec.version>
  <library-maven-plugin-compiler.version>3.8.0</library-maven-plugin-
  compiler.version>
  <library-jaxws-ri.version>2.3.0</library-jaxws-ri.version>
  <library-cometd.version>3.1.5</library-cometd.version>
</properties>

  <dependencies>
    <dependency>
      <groupId>javax.xml.ws</groupId>
      <artifactId>jaxws-api</artifactId>
      <version>${library-jaxws-ri.version}</version>
    </dependency>
    <dependency>
      <groupId>esa.egos.edds</groupId>
      <artifactId>edds-ws-client</artifactId>
      <version>${edds.version}</version>
    </dependency>
    <dependency>
      <groupId>esa.egos.edds</groupId>
      <artifactId>edds-ws-client</artifactId>
      <version>${edds.version}</version>
      <classifier>tests</classifier>
    </dependency>
```

```
<dependency>
  <groupId>esa.egos.edds</groupId>
  <artifactId>edds-ws-common</artifactId>
  <version>${edds.version}</version>
</dependency>
<dependency>
  <groupId>esa.egos.edds</groupId>
  <artifactId>edds-common</artifactId>
  <version>${edds.version}</version>
</dependency>
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>${library-commons-codec.version}</version>
</dependency>
<dependency>
  <groupId>org.cometd.java</groupId>
  <artifactId>bayeux-api</artifactId>
  <version>${library-cometd.version}</version>
</dependency>
<dependency>
  <groupId>org.eclipse.jetty</groupId>
  <artifactId>jetty-client</artifactId>
  <version>${library-jetty.version}</version>
</dependency>
<dependency>
<groupId>org.eclipse.jetty</groupId>
  <artifactId>jetty-util-ajax</artifactId>
  <version>${library-jetty.version}</version>
</dependency>
<dependency>
  <groupId>org.cometd.java</groupId>
  <artifactId>cometd-java-client</artifactId>
  <version>${library-cometd.version}</version>
</dependency>
<dependency>
  <groupId>com.google.protobuf</groupId>
  <artifactId>protobuf-java</artifactId>
  <version>${library-protobuf-java.version}</version>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>${library-maven-plugin-compiler.version}</version>
  <configuration>
    <release>11</release>
  </configuration>
</plugin>
</plugins>
</build>
```

At this stage, the project settings are finished and it is possible to add client code. In this paragraph two complete listings are offered:

- ***ParameterDefinitionDownloader***, provided in 12.1.9.2, a Java Application which submits a Parameter Definition Batch Request to EDDS and downloads the response when it is ready
- ***DARCPParameterDownloader***, provided in 12.1.9.3, a Java Application which submits a DARC Parameter Batch Request to EDDS and downloads the response when it is ready.

The two applications need some values to be tuned, such as:

- ***EDDS EndPoint***, containing EDDS URL, such as for example <http://10.36.97.16:8080/edds/EddsService?wsdl>
- EDDS ***username*** and ***password***;
- ***Mission***, ***domain*** and ***rule*** for the Batch request context
- The ***path*** of the local download directory

The applications can be run as simple java applications.

12.1.9.2 ParameterDefinitionDownloader

Example provided in Appendix ***O.1***

12.1.9.3 DARCPParameterDownloader

Example provided in Appendix ***O.2***

12.2 EDDS Push notifications client

EDDS clients can register their listeners to receive push notification on messages from EDDS system. The client listeners are called asynchronously when any messages are received. Supported notification types are:

Channel (from enum NotificationChannel)	Return Type	Description
USER_MANAGEMENT	AccountRequestMessagePart	User management updates
STATUS_UPDATES	RequestInfoRec	Job updates
STATUS_RETRIEVAL	RequestInfoRecs	Used to get a list of jobs and their status
STATUS_RETRIEVAL_ERRORS	String	Error message if unable to retrieve status information
EDDS_LOGS	LoggingEvent	EDDS live system log messages
PARAM_DEFINITIONS	ParamDefRecs	Parameter definitions in response to the requests through the web service.
PARAM_DEFINITIONS_ERRORS	String	Parameter definition errors
ARCHIVE_CATALOGUE		The FARC catalogue entries
ARCHIVE_CATALOGUE_ERRORS	String	Any error information when retrieving the catalogue
ARCHIVE_UPDATES	FarcCatRec	Information about a new commit in the FARC
HISTORICAL_LOGS	SystemLogRec	Historical logs in response to the requests through the web service.
USER_QUOTA_USAGE	UserQuotaDetailsList	Quota updates
PARAM_STREAM	DarcParamRec	DARC parameter stream
PKT_TC_STREAM	PacketTCBinary	Telecommand stream
PKT_TM_STREAM	PacketTMBinary	Telemetry stream
PKT_EV_STREAM	EventRecordReportBinary	SCOS Event Log stream
PKT_OOL_STREAM	OOLReportBinary	Out of limits stream
FILE_SYSTEM_CATALOGUE	Catalogue	The File System Catalogue entries in response to the Special request for File System
PKT_CANCEL_STREAM	String	Used to notify stream client request ids cancelled or completed.

Table 38 - EDDS Push notifications client

12.2.1 Security

The same session key that is obtained during login of EDDS Web services is used to authenticate the client. If client fails to provide the server with an existing session key, the service is denied.

12.2.2 Example: Register notifications listener

```
/**
 * Register your listener to receive asynchronous notifications for
status updates
 *
 * @param listener the listener the will receive updates when they
are received from the EDDS Web Server
 * @see NotificationManager#registerListener(NotificationChannel,
NotificationListener) for the list supported types
 */
public void registerStatusUpdatesListener(NotificationListener
<RequestInfoRec> listener)
{
    notificationManager.
registerListener(NotificationChannel.STATUS_UPDATES, listener);
}
```

12.3 EDDS Utility methods

12.3.1 Decoding EDDS Param Binary format

EDDS Client API provides utility functions to read and write the TM Parameter binary format. The implementation is in package `esa.egos.edds.ws.client.binary`. For more detailed documentation read the Javadoc of the provided classes.

12.3.1.1 Example: Reading Parameter TM data from EDDS Binary file

```
/**
 * Read the Parameter BINARY response file. It contains a list of {@link
DarcParameterBinary} objects
 * encoded using Google Protocol Buffers. Each record is prepended with
it's length.
 * @param file the file object of the file being read.
 * @throws InvalidProtocolBufferException when something unexpected occurs
while reading the data
 * @throws IOException problem reading the file
 */
public void readParamBinaryFormat(File file) throws
InvalidProtocolBufferException, IOException{
    DarcParameterBinaryInputSupport bin = null;
    try
    {
        bin = new DarcParameterBinaryInputSupport(file);
        DarcParameterBinary rec;
        while ( (rec = bin.next()) != null)
        {
            System.out.println(rec.getParamName() + " " + rec.getGenTime()
+ ": " + rec.getValue());
        }
    }
    finally
    {
        if (bin != null)
        {
            bin.close();
        }
    }
}
```

12.4 Dependencies

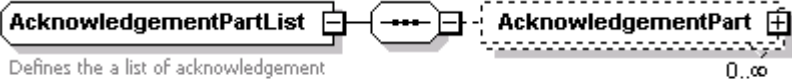
The EDDS Client API can be used by including the respective Maven artifact to the project. Maven then will resolve the dependencies and add all required libraries to the classpath. To use the EDDS Client API add the following Maven artifact to the project:

```
<dependency>
  <groupId>esa.egos.edds</groupId>
  <artifactId>edds-ws-client</artifactId>
  <version>3.6.0</version>
</dependency>
```

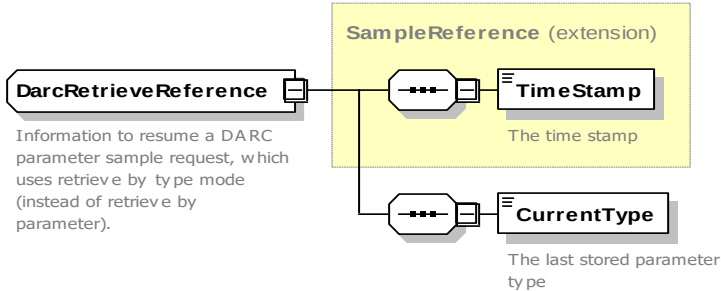
For a complete list of dependencies, please check the Table 37.

namespace	http://edds.egos.esa/model
children	ContextPart RequestId State Retries Reason ProblemRecords Error ReportCreationDate ReportCreationDOYDate SubmissionTime SubmissionDOYTime ScheduledTime ScheduledDOYTime StartTime StartDOYTime CompletionTime CompletionDOYTime ExpiryTime ExpiryDOYTime Priority Percentage RetrievedSamples FilteredSamples User ResponseFiles AckDelivery Delivery FilterTimeWindow LastSampleReference
used by	element AcknowledgementPartList/AcknowledgementPart
annotation	documentation Defines the acknowledgement message created at the different stages of the request execution

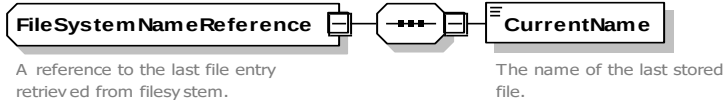
A.2 *complexType AcknowledgementPartList*

diagram	
namespace	http://edds.egos.esa/model
children	AcknowledgementPart
annotation	documentation Defines a list of acknowledgement messages

A.3 *complexType DarcRetrieveReference*

diagram	
namespace	http://edds.egos.esa/model
type	extension of SampleReference
properties	base SampleReference
children	TimeStamp CurrentType
annotation	documentation Information to resume a DARC parameter sample request, which uses retrieve by type mode (instead of retrieve by parameter).

A.4 *complexType FileSystemNameReference*

diagram	
namespace	http://edds.egos.esa/model
type	extension of RetrievalReference
properties	base RetrievalReference
children	CurrentName

annotation	documentation A reference to the last file entry retrieved from filesystem.
------------	--

A.5 complexType ParameterSampleReference

diagram	
namespace	http://edds.egos.esa/model
type	extension of SampleReference
properties	base SampleReference
children	TimeStamp ParamName
annotation	documentation References a parameter sample. Holds a parameter name and a timestamp.

A.6 complexType ResponseFileEntry

diagram	
namespace	http://edds.egos.esa/model
children	FileName Size Status Error Checksum
used by	element AcknowledgementPart/ResponseFiles
annotation	documentation Defines the response file and it's status

A.7 complexType RetrievalReference

diagram	RetrievalReference A bstract ty pe that defines a reference to the last stored location of data. This is the parent for all other reference ty pes.
namespace	http://edds.egos.esa/model
properties	abstract true
used by	element AcknowledgementPart/LastSampleReference complexTypes FileSystemNameReference SampleReference
annotation	documentation Abstract type that defines a reference to the last stored location of data. This is the parent for all other reference types.

A.8 complexType SampleReference

diagram	SampleReference Defines a reference to a response sample. Time Stamp The time stamp
namespace	http://edds.egos.esa/model
type	extension of RetrievalReference
properties	base RetrievalReference
children	TimeStamp
used by	complexTypes DarcRetrieveReference ParameterSampleReference SampleTimeStamp
annotation	documentation Defines a reference to a response sample.

A.9 complexType SampleTimeStamp

diagram	SampleTimeStamp References a sample by its time stamp. SampleReference (extension) Time Stamp The time stamp
namespace	http://edds.egos.esa/model
type	extension of SampleReference
properties	base SampleReference
children	TimeStamp
annotation	documentation References a sample by its time stamp.

A.10 simpleType ResponseFileStatus

namespace	http://edds.egos.esa/model
-----------	----------------------------

type	restriction of xs:string		
properties	base	xs:string	
used by	element	ResponseFileEntry/Status	
facets	Kind enumeration	Value ACTIVE	annotation documentation The file is being written by the EDDS Server
	enumeration	SERVER_COMPLETED	documentation EDDS Server completed writing the file, delivery- manager needs to deliver the file
	enumeration	DELIVERED_LOCAL	documentation The file is available in the EDDS Server FTP directory
	enumeration	DELIVERED_LOCAL_REMOTE	documentation The file is delivered to the remote server and is also available in the EDDS Server FTP directory
	enumeration	DELIVERED_REMOTE	documentation The delivery of the file is completed
	enumeration	DELETED	documentation The data on EDDS file server is deleted. Can be reached after DELIVERED_LOCALLY or DELIVERED_LOCAL_REMOTE
	enumeration	ERROR	documentation The file cannot be saved on EDDS file server
annotation	documentation Response file statuses		

Appendix B Authentication Data Type

This section of the document describes the complex types used for authentication as defined in the authentication.xsd file.

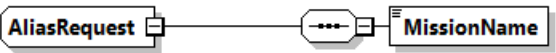
B.1 complexType UserCredentials

diagram	
namespace	http://edds.egos.esa/model
children	UserName Password SessionId
used by	element UserCredentials
annotation	documentation Security info needed during the user login

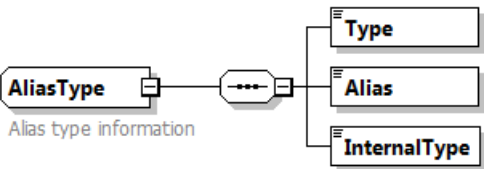
Appendix C Batch Request Data Type

This section of the document describes the complex types used for batch requests as defined in the batchrequest.xsd file.

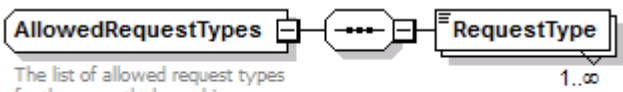
C.1 *complexType AliasRequest*

diagram	 The request for alias for given mission
namespace	http://edds.egos.esa/model
children	string
used by	
annotation	documentation The request for alias for given mission

C.2 *complexType AliasType*

diagram	 Alias type information
namespace	http://edds.egos.esa/model
children	string string AliasInternalType
used by	ListAliasInfo
annotation	documentation Alias type information

C.3 *complexType AllowedRequestTypes*

diagram	 The list of allowed request types for the currently logged-in user 1..∞
namespace	http://edds.egos.esa/model
children	RequestType
used by	element AllowedRequestTypes
annotation	documentation The list of allowed request types for the currently logged-in user

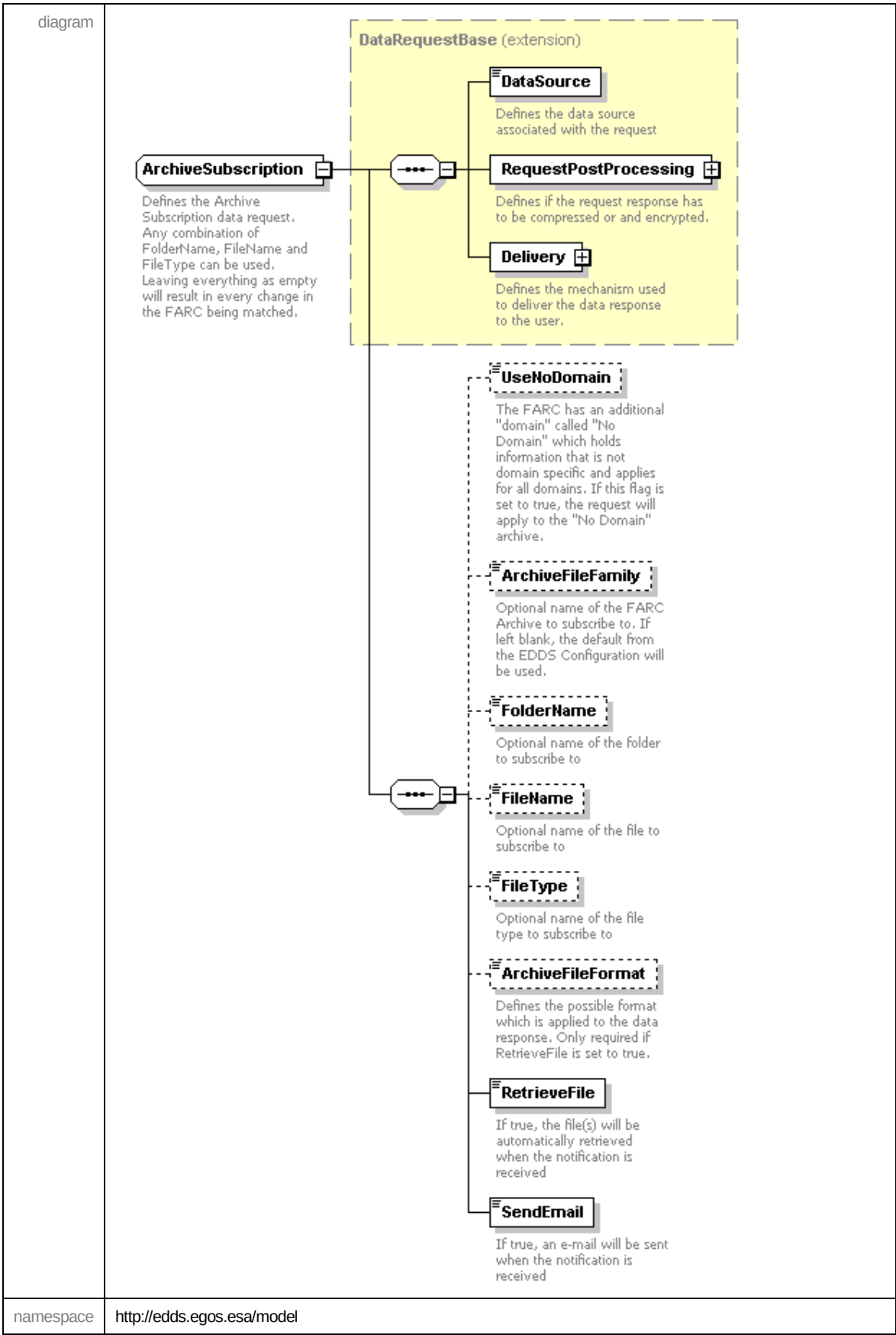
C.4 complexType ArchiveCatalogue

diagram	ArchiveCatalogue Defines the Archive Catalogue data request DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. UseNoDomain The FARC has an additional "domain" called "No Domain" which holds information that is not domain specific and applies for all domains. If this flag is set to true, the request will apply to the "No Domain" archive. CatalogueFilter Defines the basic information necessary to retrieve the data Catalogue from the FARC archive and defines the filter to be applied to the retrieved data. CatalogueFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery UseNoDomain CatalogueFilter CatalogueFormat
used by	elements ArchiveCatalogue DataRequest/ArchiveCatalogue
annotation	documentation Defines the Archive Catalogue data request

C.5 complexType ArchiveFile

diagram	The diagram illustrates the structure of the ArchiveFile complexType. It is an extension of the DataRequestBase type. DataRequestBase (highlighted in yellow) contains three elements: DataSource (Defines the data source associated with the request), RequestPostProcessing (Defines if the request response has to be compressed or and encrypted), and Delivery (Defines the mechanism used to deliver the data response to the user). The ArchiveFile type (Defines the Archive File data request) extends DataRequestBase and includes three additional elements: UseNoDomain (The FARC has an additional "domain" called "No Domain" which holds information that is not domain specific and applies for all domains. If this flag is set to true, the request will apply to the "No Domain" archive), ArchiveFileFilter (Defines the basic information necessary to retrieve the data file from the FARC archive and defines the filter to be applied to the retrieved data), and ArchiveFileFormat (Defines the possible format which is applied to the data response).
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery UseNoDomain ArchiveFileFilter ArchiveFileFormat
used by	elements ArchiveFile DataRequest/ArchiveFile
annotation	documentation Defines the Archive File data request

C.6 complexType ArchiveSubscription



type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery UseNoDomain ArchiveFileFamily FolderName FileName FileType ArchiveFileFormat RetrieveFile SendEmail
used by	elements ArchiveSubscription DataRequest/ArchiveSubscription
annotation	documentation Defines the Archive Subscription data request. Any combination of FolderName, FileName and FileType can be used. Leaving everything as empty will result in every change in the FARC being matched.

C.7 *complexType* **ArchiveSubscriptionNotification**

diagram	<pre>classDiagram class ArchiveSubscriptionNotification { RequestId FolderName FileName FileType Domain ArchiveFamily OperationType Issue Revision }</pre> The diagram shows the ArchiveSubscriptionNotification class structure. It includes the following attributes and their descriptions: RequestId: The ID of the request FolderName: Name of the folder that changed FileName: Name of the file that changed FileType: Type of the file that changed Domain: The domain in which the file changed ArchiveFamily: The archive family in which the file changed OperationType: The operation performed on the file Issue: The issue number of the changed file Revision: The revision number of the changed file
namespace	http://edds.egos.esa/model
children	RequestId FolderName FileName FileType Domain ArchiveFamily OperationType Issue Revision
used by	element ArchiveSubscriptionNotification
annotation	documentation Information about a commit to the FARC

C.8 complexType BatchRequest

diagram	
namespace	http://edds.egos.esa/model
children	Comment Schedule User DataRequest AckDelivery
used by	elements BatchRequest RequestMessagePart/BatchRequest
annotation	documentation Defines the structure and content of a batch request.

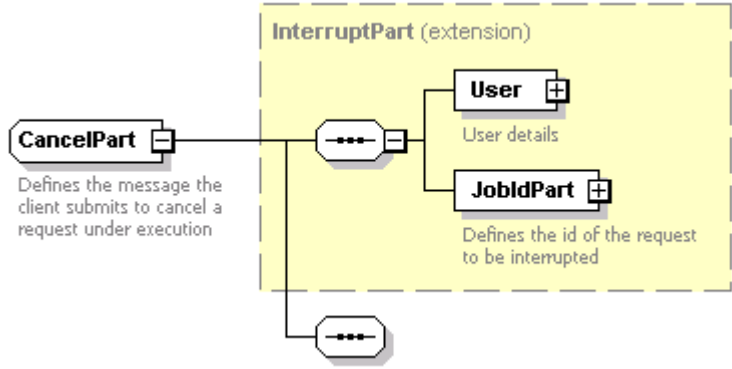
C.9 complexType BehaviourState

diagram	
namespace	http://edds.egos.esa/model
children	StatusType OOLState DetailedState NrOfViolations Value
used by	element BehaviourStates/BehaviourState
annotation	documentation Behaviour state

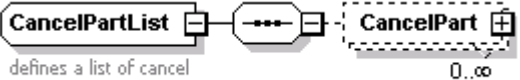
C.10 complexType BehaviourStates

diagram	
namespace	http://edds.egos.esa/model
children	BehaviourState
used by	element ParamRepresentationSample/BehaviourStates
annotation	documentation Behaviour states

C.11 complexType CancelPart

diagram	
namespace	http://edds.egos.esa/model
type	extension of InterruptPart
properties	base InterruptPart
children	User JobIdPart
used by	element CancelPart
annotation	documentation Defines the message the client submits to cancel a request under execution

C.12 complexType CancelPartList

diagram	
namespace	http://edds.egos.esa/model
children	CancelPart
used by	element CancelPartList
annotation	documentation defines a list of cancel requests

C.13 complexType CatalogueRequest

diagram	The request for the catalogue for given data source, mission, domain and folder. If the folder is specified the request will be non-recursive.
namespace	http://edds.egos.esa/model
children	DataSource MissionName Domain Folder
used by	element CatalogueRequest
annotation	documentation The request for the catalogue for given data source, mission, domain and folder. If the folder is specified the request will be non-recursive.

C.14 complexType CatalogueResponse

diagram	Defines the list of catalogue entry 0..∞ Defines the single catalogue entry
namespace	http://edds.egos.esa/model
children	CatalogueResponseElement
used by	element Response/CatalogueResponse
annotation	documentation Defines the list of catalogue entry

C.15 complexType CatalogueResponseElement

diagram	CatalogueResponseElement Describes the single catalogue entry DataObjectId Defines the unique object id FileName Defines the name of the catalogue file entry FileType Defines the type of the catalogue file entry CreationDate Defines the creation type of the file Release Defines the release id of the catalogue file entry Issue Defines the issue id of the catalogue file entry Description File description Comment File comment
namespace	http://edds.egos.esa/model
children	DataObjectId FileName FileType CreationDate Release Issue Description Comment
used by	element CatalogueResponse/CatalogueResponseElement
annotation	documentation Describes the single catalogue entry

C.16 complexType DataRequest

diagram	<pre> graph TD DataRequest[DataRequest] --> Param[Param] DataRequest --> EventRecordReport[EventRecordReport] Param --> ParamStatistics[ParamStatistics] Param --> ParamDefinition[ParamDefinition] Param --> ParamPreview[ParamPreview] Param --> SmonParam[SmonParam] Param --> PktEv[PktEv] Param --> PktTm[PktTm] Param --> PktTc[PktTc] Param --> PktTmStatistics[PktTmStatistics] Param --> PktTcStatistics[PktTcStatistics] Param --> PktEvStatistics[PktEvStatistics] Param --> PktTmReport[PktTmReport] Param --> PktTcReport[PktTcReport] Param --> PktEvRaw[PktEvRaw] Param --> PktTmRaw[PktTmRaw] Param --> PktTcRaw[PktTcRaw] Param --> PktTmGapReport[PktTmGapReport] Param --> ArchiveCatalogue[ArchiveCatalogue] Param --> ArchiveFile[ArchiveFile] Param --> ArchiveSubscription[ArchiveSubscription] Param --> OolRecordReport[OolRecordReport] Param --> EdsUsageReport[EdsUsageReport] Param --> FileSystemFileCatalogue[FileSystemFileCatalogue] Param --> FileSystemFolderCatalogue[FileSystemFolderCatalogue] Param --> FileSystemFile[FileSystemFile] Param --> FileSystemSubscription[FileSystemSubscription] </pre> DataRequest Defines the specific data requested in the BatchRequest Param Defines the Parameter Batch Request ParamStatistics Defines the Parameter Statistics Batch Request ParamDefinition Defines the Parameter Definition Batch Request ParamPreview Defines the Parameter Preview Batch Request SmonParam Defines the SMON Parameter Batch Request PktEv Defines the Event Packet batch request. PktTm Defines the Telemetry Packet batch request. PktTc Defines the Telecommand Packet batch request. PktTmStatistics Defines the Packet Tm Statistics batch request. PktTcStatistics Defines the Packet Tc Statistics batch request. PktEvStatistics Defines the Packet Ev Statistics batch request. PktTmReport Defines the Pkt Tm Mcs Report batch request PktTcReport Defines the Pkt Tc Mcs Report batch request EventRecordReport Defines the event record report PktEvRaw Defines the Event Packet batch request. PktTmRaw Defines the Telemetry Packet batch request. PktTcRaw Defines the Telecommand Packet batch request. PktTmGapReport Defines the Telemetry Gaps Report batch request. ArchiveCatalogue Defines the Archive Catalogue batch request ArchiveFile Defines the File batch request ArchiveSubscription Defines the File subscription request OolRecordReport Defines the Out Of Limit Report Batch Request EdsUsageReport Defines the Eds Usage Report Request FileSystemFileCatalogue Request to non-recursively list all the files in the specified directories FileSystemFolderCatalogue Request to recursively list all the subfolders of the specified directories FileSystemFile Request the files FileSystemSubscription Request the files when they are created or modified.
namespace	http://eds.egos.esa/model
children	Param ParamStatistics ParamDefinition ParamPreview SmonParam PktEv PktTm PktTc PktTmStatistics PktTcStatistics PktEvStatistics PktTmReport PktTcReport EventRecordReport PktEvRaw PktTmRaw PktTcRaw PktTmGapReport ArchiveCatalogue ArchiveFile ArchiveSubscription OolRecordReport EdsUsageReport FileSystemFileCatalogue FileSystemFolderCatalogue FileSystemFile FileSystemSubscription

	FileSystemFolderCatalogue FileSystemFile FileSystemSubscription
used by	elements DataRequest BatchRequest/DataRequest
annotation	documentation Defines the specific data requested in the BatchRequest

C.17 complexType DataspacesRequest

diagram	The request for available dataspaces for given mission, domain (optional) and request type
namespace	http://edds.egos.esa/model
children	MissionName Domain DataSource
used by	element DataspacesRequest
annotation	documentation The request for available dataspaces for given mission, domain (optional) and request type

C.18 complexType DefaultFileServerRequest

diagram	The request for the default file server for the specified mission and role
namespace	http://edds.egos.esa/model
children	MissionName RoleName
used by	element DefaultFileServerRequest
annotation	documentation The request for the default file server for the specified mission and role

C.19 complexType DetailedBehaviourState

diagram	Detailed state
---------	-----------------------

namespace	http://edds.egos.esa/model
children	LimitState StatusState SCCState DeltaState
used by	element BehaviourState/DetailedState
annotation	documentation Detailed state

C.20 complexType DownloadList

diagram	Defines the job id plus the list of files to download. If the list of files to download is empty, all files will be returned.
namespace	http://edds.egos.esa/model
children	Job FileList
used by	element DownloadList

C.21 complexType EgsgccActivityReport

diagram	
namespace	http://edds.egos.esa/model
children	
used by	element

C.22 complexType EgsgccActivityReportListElement

diagram	
namespace	http://edds.egos.esa/model
children	
used by	element

C.23 complexType FarcCatalogueRequest

diagram	The request for the FARC catalogue for given mission, domain and folder. If the folder is specified the request will be non-recursiv e. @deprecated, use CatalogueRequest
---------	--

namespace	http://edds.egos.esa/model
children	MissionName Domain Folder
used by	element FarcCatalogueRequest
annotation	documentation The request for the FARC catalogue for given mission, domain and folder. If the folder is specified the request will be non-recursive. @deprecated, use CatalogueRequest

C.24 complexType InterruptPart

diagram	Defines the message the client submits to interrupt a request under execution. Is not meant to be used directly; see child objects CancelPart and SuspendPart instead.
namespace	http://edds.egos.esa/model
children	User JobIdPart
used by	complexType CancelPart ResumePart SuspendPart
annotation	documentation Defines the message the client submits to interrupt a request under execution. Is not meant to be used directly; see child objects CancelPart and SuspendPart instead.

C.25 complexType JobsFilter

diagram	JobsFilter Defines the batch request message the client submits to server TimeWindow Defines the time range of the request. FilterType Whether the extended filters should be applied with an AND OR condition. RequestType 0..∞ SubType 0..∞ Status 0..∞ DomainId 0..∞ MissionId 0..∞ UserName 0..∞ Limit Defines the limit. Direction Defines whether to retrieve ascending or descending. If not set, the default is descending.
namespace	http://edds.egos.esa/model
children	TimeWindow Limit Direction
used by	
annotation	documentation Defines the batch request message the client submits to server

C.26 complexType JobIdPart

diagram	JobIdPart Defines the job is plus the job context Job Defines the jobId
namespace	http://edds.egos.esa/model
children	Job
used by	elements DeleteDataPart DeleteRequestPart JobIdPart CancelPart/JobIdPart
annotation	documentation Defines the job is plus the job context

C.27 complexType JobIdPartList

diagram	Defines a list of jobs 0..∞
namespace	http://edds.egos.esa/model
children	JobIdPart
used by	elements DeleteDataPart DeleteRequestPart JobIdPartList
annotation	documentation Defines a list of jobs

C.28 complexType JobListIdPart

diagram	JobId list 0..∞ Job Id
namespace	http://edds.egos.esa/model
children	JobListIdPartElement
used by	element JobListIdPart
annotation	documentation JobId list

C.29 complexType LastConsolidationRequest

diagram	The request for last DARC consolidation time for given dataspace, or active one, if none specified
namespace	http://edds.egos.esa/model
children	MissionName DataSource Dataspace
used by	element LastConsolidationRequest
annotation	documentation The request for last DARC consolidation time for given dataspace, or active one, if none specified

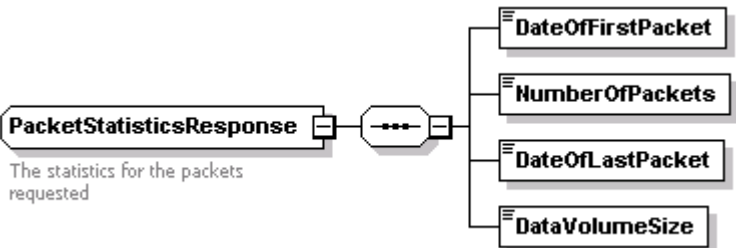
C.30 complexType ListAliasInfo

diagram	A list of AliasInfo 0..∞
namespace	http://edds.egos.esa/model
children	AliasType
used by	
annotation	documentation A list of AliasInfo

C.31 complexType ListString

diagram	
namespace	http://edds.egos.esa/model
children	ListElement
used by	element ListString
annotation	documentation A list of String values

C.32 complexType PacketStatisticsResponse

diagram	
namespace	http://edds.egos.esa/model
children	DateOfFirstPacket NumberOfPackets DateOfLastPacket DataVolumeSize
used by	element Response/PacketStatisticsResponse
annotation	documentation The statistics for the packets requested

C.33 complexType Param

diagram	The diagram illustrates the structure of the Param complex type. It is an extension of DataRequestBase. The Param type has a required role (indicated by a small square) that connects to a provided role (indicated by a small circle with three dots) within DataRequestBase. This provided role is further connected to three other roles: DataSource, RequestPostProcessing, and Delivery. Additionally, Param has a provided role that connects to ParamTmFilter and ParamTmFormat. Param (Required Role): Defines the Parameter data request DataRequestBase (extension) DataSource (Provided Role): Defines the data source associated with the request RequestPostProcessing (Provided Role): Defines if the request response has to be compressed or and encrypted. Delivery (Provided Role): Defines the mechanism used to deliver the data response to the user. ParamTmFilter (Provided Role): Defines the basic information necessary to retrieve the Parameter data from the DARC archive and defines the filter to be applied to the retrieved data. ParamTmFormat (Provided Role): Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery ParamTmFilter ParamTmFormat
used by	elements Param DataRequest/Param
annotation	documentation Defines the Parameter data request

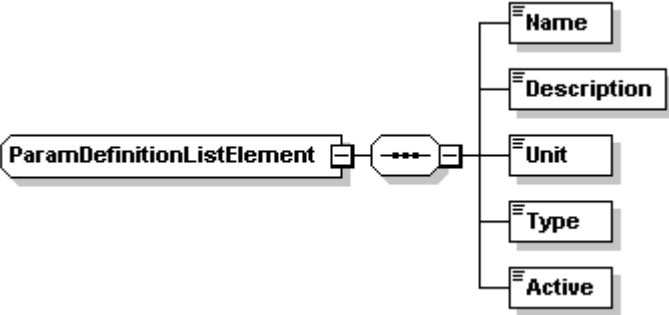
C.34 complexType ParamDefinition

diagram	The diagram illustrates the structure of the ParamDefinition complex type. It is an extension of DataRequestBase. The ParamDefinition type contains three children: DataSource, RequestPostProcessing, and Delivery, which are grouped within a dashed box labeled DataRequestBase (extension). Additionally, ParamDefinition has a child ParamDefinitionFormat. The descriptions for each element are as follows: ParamDefinition: Defines the Parameter Definition data request DataSource: Defines the data source associated with the request RequestPostProcessing: Defines if the request response has to be compressed or and encrypted. Delivery: Defines the mechanism used to deliver the data response to the user. ParamDefinitionFormat: Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery ParamDefinitionFormat
used by	elements ParamDefinition DataRequest/ParamDefinition
annotation	documentation Defines the Parameter Definition data request

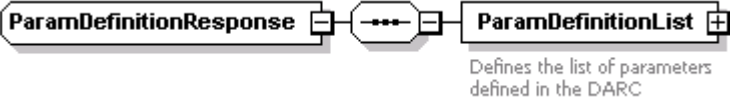
C.35 complexType ParamDefinitionList

diagram	The diagram illustrates the structure of the ParamDefinitionList complex type. It contains a single child ParamDefinitionListElement. The diagram uses standard UML notation for complex types and their children.
namespace	http://edds.egos.esa/model
children	ParamDefinitionListElement
used by	element ParamDefinitionResponse/ParamDefinitionList

C.36 complexType ParamDefinitionListElement

diagram	 The diagram shows the structure of the ParamDefinitionListElement complex type. It consists of a root element 'ParamDefinitionListElement' (represented by a rounded rectangle with a small square on the right) connected to a sequence container (a rectangle with three dots inside). This container is connected to five child elements: 'Name', 'Description', 'Unit', 'Type', and 'Active', each represented by a rectangle with a small square on the right.
namespace	http://edds.egos.esa/model
children	Name Description Unit Type Active
used by	elements ParamDefinitionListElement ParamDefinitionList/ParamDefinitionListElement

C.37 complexType ParamDefinitionResponse

diagram	 The diagram shows the structure of the ParamDefinitionResponse complex type. It consists of a root element 'ParamDefinitionResponse' (represented by a rounded rectangle with a small square on the right) connected to a sequence container (a rectangle with three dots inside), which is then connected to the 'ParamDefinitionList' element (represented by a rectangle with a small square on the right). Below the diagram, the text 'Defines the list of parameters defined in the DARC' is present.
namespace	http://edds.egos.esa/model
children	ParamDefinitionList
used by	element Response/ParamDefinitionResponse

C.38 complexType ParamPacketInstance

diagram	
namespace	http://edds.egos.esa/model
children	ReceptionTime GenerationTime ReceptionTimeDOY GenerationTimeDOY SampleOffset SequenceCounter SpacecraftID Validity DataStream Domain PacketSize GroundStation PacketMnemonic
used by	elements ParamPacketInstances/PacketInstance ParamPacketInstance
annotation	documentation Parameter Packet Instances for Parameters from SMON

C.39 complexType ParamPacketInstances

diagram	Parameter Packet Instances for Parameters from SMON 0..∞ Defines the parameter sample element
namespace	http://edds.egos.esa/model
children	PacketInstance
used by	element SMONParamSampleListElement/ParamPacketInstances
annotation	documentation Parameter Packet Instances for Parameters from SMON

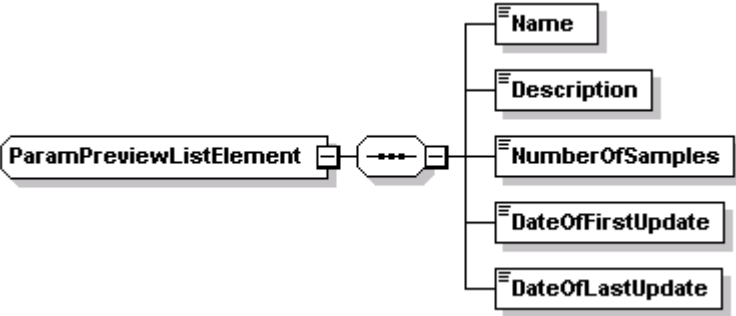
C.40 complexType ParamPreview

diagram	
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery ParamPreviewFilter ParamPreviewFormat
used by	elements ParamPreview DataRequest/ParamPreview
annotation	documentation Defines the Parameter Preview data request

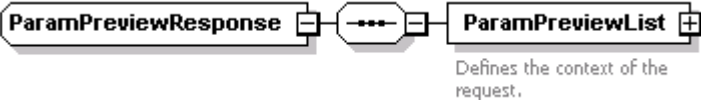
C.41 complexType ParamPreviewList

diagram	 The diagram shows a box labeled 'ParamPreviewList' connected to a dashed box containing three dots, which is then connected to a box labeled 'ParamPreviewListElement'. Below the dashed box is the cardinality '0..∞' and the text 'ParameterDefinition element'.
namespace	http://edds.egos.esa/model
children	ParamPreviewListElement
used by	element ParamPreviewResponse/ParamPreviewList

C.42 complexType ParamPreviewListElement

diagram	 The diagram shows a box labeled 'ParamPreviewListElement' connected to a dashed box containing three dots, which is then connected to a vertical stack of five boxes: 'Name', 'Description', 'NumberOfSamples', 'DateOfFirstUpdate', and 'DateOfLastUpdate'.
namespace	http://edds.egos.esa/model
children	Name Description NumberOfSamples DateOfFirstUpdate DateOfLastUpdate
used by	elements ParamPreviewListElement ParamPreviewList/ParamPreviewListElement

C.43 complexType ParamPreviewResponse

diagram	 The diagram shows a box labeled 'ParamPreviewResponse' connected to a dashed box containing three dots, which is then connected to a box labeled 'ParamPreviewList'. Below the dashed box is the text 'Defines the context of the request.'.
namespace	http://edds.egos.esa/model
children	ParamPreviewList
used by	element Response/ParamPreviewResponse

C.44 complexType ParamRepresentationSample

diagram	
namespace	http://edds.egos.esa/model
children	RepresentationName SampleTime SampleTimeDOY Value Validity BehaviourStates
used by	elements ParamRepresentationSample ParamRepresentationSamples/ParamRepresentationSample
annotation	documentation Parameter representation sample

C.45 complexType ParamRepresentationSamples

diagram	
namespace	http://edds.egos.esa/model
children	ParamRepresentationSample
used by	element SMONParamSampleListElement/ParamRepresentationSamples
annotation	documentation Parameter representation samples from SMON

C.46 complexType ParamResponse

diagram	
namespace	http://edds.egos.esa/model
children	ParamSampleList MinimumThreshold MaximumThreshold User RetrievalStart RetrievalStartDOY RetrievalStop RetrievalStopDOY
used by	element Response/ParamResponse
annotation	documentation Defines the XML response returned on Parameter Batch Requests

C.47 complexType ParamSampleList

diagram	
namespace	http://edds.egos.esa/model
children	ParamSampleListElement

used by	element ParamResponse/ParamSampleList
annotation	documentation Defines the list of parameter samples

C.48 complexType ParamSampleListElement

diagram	TimeStampAsciiA Parameter time stamp as CCSDS AsciiA (yyyy-mm-ddThh:mm:ss.mmm) TimeStampAsciiB Parameter time stamp as CCSDS AsciiB (yyyy-dddThh:mm:ss.mmm) StorageTimeStampAsciiA Time that the parameter sample is stored in the DARC (time written to the buffer) as CCSDS AsciiA (yyyy-mm-ddThh:mm:ss.mmm). Not present when saving a DARC parameter stream. StorageTimeStampAsciiB Time that the parameter sample is stored in the DARC (time written to the buffer) as CCSDS AsciiB (yyyy-dddThh:mm:ss.mmm). Not present when saving a DARC parameter stream. Name Parameter Name Description Parameter Description Unit Parameter Engineering unit Validity Parameter Engineering validity EngineeringValue Parameter Eng Value Type Parameter Type ParentId The parent ID of the TM parameter sample. For SCOS sources, this would be the SPID ParentGenTimeAsciiA The parent generation time as CCSDS AsciiA (yyyy-mm-ddThh:mm:ss.mmm). For SCOS sources, this would be the packet generation time ParentGenTimeAsciiB The parent generation time as CCSDS AsciiB (yyyy-dddThh:mm:ss.mmm). For SCOS sources, this would be the packet generation time RawValue Parameter Raw Value RawValueType The type of the raw value
namespace	http://edds.egos.esa/model
children	TimeStampAsciiA TimeStampAsciiB StorageTimeStampAsciiA StorageTimeStampAsciiB Name Description Unit Validity EngineeringValue Type ParentId ParentGenTimeAsciiA ParentGenTimeAsciiB RawValue RawValueType

	RawValue RawValueType
used by	elements ParamSampleListElement ParamSampleList/ParamSampleListElement

C.49 complexType ParamStatisticListElement

diagram						
namespace	http://edds.egos.esa/model					
children	StatisticsRecord					
used by	element	ParamStatisticsList/ParamStatisticListElement				
attributes	Name	Type	Use	Default	Fixed	annotation
	name	xs:string				
annotation	documentation A list of statistics for a particular parameter					

C.50 complexType ParamStatistics

diagram						
namespace	http://edds.egos.esa/model					
type	extension of DataRequestBase					
properties	base	DataRequestBase				

children	RequestPostProcessing Delivery ParamStatisticsFilter ParamStatisticsFormat
used by	elements ParamStatistics DataRequest/ParamStatistics
annotation	documentation Defines the Parameter Statistics data request

C.51 complexType ParamStatisticsList

diagram	A list of parameters with all the associated statistics for the time period requested
namespace	http://edds.egos.esa/model
children	ParamStatisticListElement
used by	element ParamStatisticsResponse/ParamStatisticList
annotation	documentation A list of parameters with all the associated statistics for the time period requested

C.52 complexType ParamStatisticsResponse

diagram	Defines the list of parameters statistics as defined in the DARC
namespace	http://edds.egos.esa/model
children	ParamStatisticList
used by	element Response/ParamStatisticsResponse

C.53 complexType PktEv

diagram	PktEv Defines the Event Packet data request DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. EvPacketFilter Defines the basic information necessary to retrieve the data from the PARC archive and defines the filter to be applied to the retrieved data. PacketFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery EvPacketFilter PacketFormat
used by	elements PktEv DataRequest/PktEv
annotation	documentation Defines the Event Packet data request

C.54 complexType PktEvRaw

diagram	PktEvRaw Defines the Event Packet raw data request, i.e. unprocessed as received from the archive. DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. RawPacketFilter Defines the basic information necessary to retrieve the data from the PARC archive PacketFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery RawPacketFilter PacketFormat
used by	elements PktEvRaw DataRequest/PktEvRaw
annotation	documentation Defines the Event Packet raw data request, i.e. unprocessed as received from the archive.

C.55 complexType PktEvStatistics

diagram	The diagram illustrates the structure of the PktEvStatistics complexType. It is an extension of DataRequestBase. The extension includes the following elements: DataSource: Defines the data source associated with the request. RequestPostProcessing: Defines if the request response has to be compressed or and encrypted. Delivery: Defines the mechanism used to deliver the data response to the user. EvPacketFilter: Defines the basic information necessary to retrieve the data from the PARC archive and defines the filter to be applied to the retrieved data. PacketStatisticsFormat: Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery EvPacketFilter PacketStatisticsFormat
used by	elements PktEvStatistics DataRequest/PktEvStatistics
annotation	documentation Defines the Event Packet statistics data request

C.56 complexType PktRawResponse

diagram	The diagram illustrates the structure of the PktRawResponse complexType. It consists of a sequence of PktRawResponseElement elements, with a cardinality of 0 to infinity (0..∞).
namespace	http://edds.egos.esa/model
children	PktRawResponseElement
used by	element Response/PktRawResponse
annotation	documentation Defines the raw packet response data

C.57 complexType PktRawResponseElement

diagram						
namespace	http://edds.egos.esa/model					
children	Packet					
used by	elements PktRawResponseElement PktRawResponse/PktRawResponseElement					
attributes	Name	Type	Use	Default	Fixed	annotation
	packetID	xs:long				
annotation	documentation Describes the single catalogue entry					

C.58 complexType PktTc

diagram						
namespace	http://edds.egos.esa/model					
type	extension of DataRequestBase					
properties	base DataRequestBase					
children	RequestPostProcessing Delivery TcPacketFilter PacketFormat					
used by	elements PktTc DataRequest/PktTc					
annotation	documentation Defines the Telecommand Packet data request					

C.59 complexType PktTcRaw

diagram	
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery RawPacketFilter PacketFormat
used by	elements PktTcRaw DataRequest/PktTcRaw
annotation	documentation Defines the Telecommand Packet raw data request, i.e. unprocessed as received from the archive.

C.60 complexType PktTcStatistics

diagram	PktTcStatistics Defines the Telecommand Statistics Packet data request DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. TcPacketFilter Defines the basic information necessary to retrieve the data from the PARC archive and defines the filter to be applied to the retrieved data. PacketStatisticsFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery TcPacketFilter PacketStatisticsFormat
used by	elements PktTcStatistics DataRequest/PktTcStatistics
annotation	documentation Defines the Telecommand Statistics Packet data request

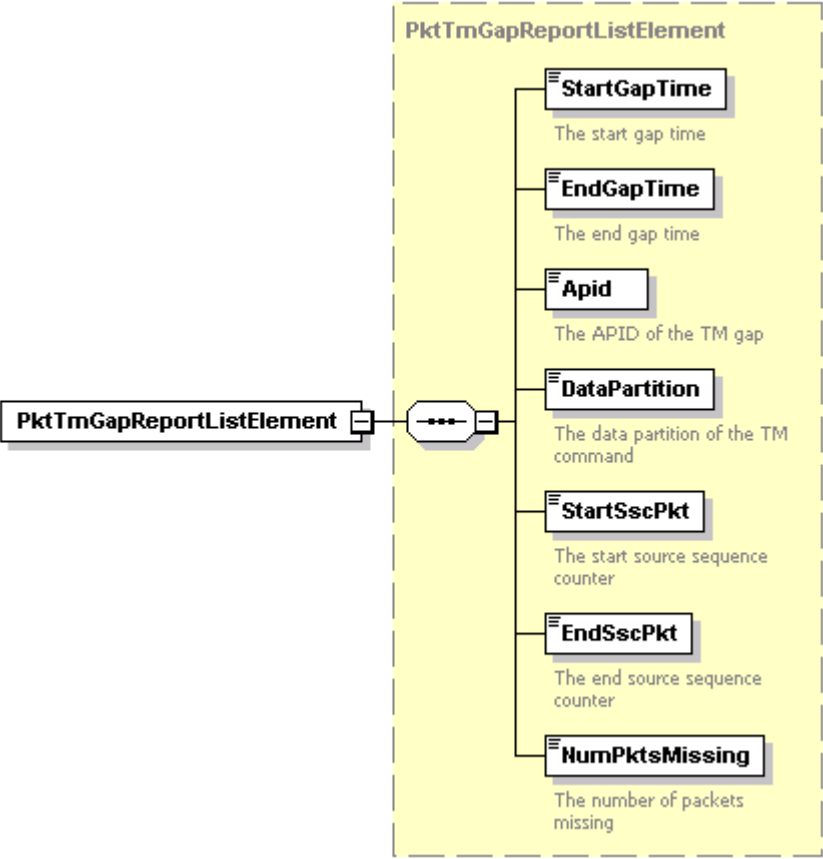
C.61 complexType PktTm

diagram	
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery TmPacketFilter PacketFormat
used by	elements PktTm DataRequest/PktTm
annotation	documentation Defines the Telemetry Packet data request

C.62 complexType PktTmGapReport

diagram	
namespace	http://edds.egos.esa/model
type	PktTmGapReport
properties	content complex
children	DataSource RequestPostProcessing Delivery PktTmGapReportFilter PktTmGapReportFormat

C.63 complexType PktTmGapReportElement

diagram	 The diagram illustrates the structure of the PktTmGapReportListElement complex type. On the left, a box labeled PktTmGapReportListElement is connected to a dashed-line box representing the element's content. This content box is titled PktTmGapReportListElement and contains a vertical sequence of seven fields, each with a label and a description: StartGapTime: The start gap time EndGapTime: The end gap time Apid: The APID of the TM gap DataPartition: The data partition of the TM command StartSscPkt: The start source sequence counter EndSscPkt: The end source sequence counter NumPktsMissing: The number of packets missing The fields are connected to a central vertical line, which is then connected to the main element box via a connector with three dots, indicating a sequence of elements.
namespace	http://edds.egos.esa/model
type	PktTmGapReportListElement
properties	content complex
children	StartGapTime EndGapTime Apid DataPartition StartSscPkt EndSscPkt NumPktsMissing

C.64 complexType PktTmRaw

diagram	The diagram illustrates the structure of the PktTmRaw complex type. It is an extension of the DataRequestBase type, which is highlighted in a yellow dashed box. PktTmRaw is connected to a choice box containing three elements: DataSource, RequestPostProcessing, and Delivery. Another choice box containing RawPacketFilter and PacketFormat is also connected to the main structure. PktTmRaw Defines the Telemetry Packet raw data request, i.e. unprocessed as received from the archive. DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. RawPacketFilter Defines the basic information necessary to retrieve the data from the PARC archive PacketFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery RawPacketFilter PacketFormat
used by	elements PktTmRaw DataRequest/PktTmRaw
annotation	documentation Defines the Telemetry Packet raw data request, i.e. unprocessed as received from the archive.

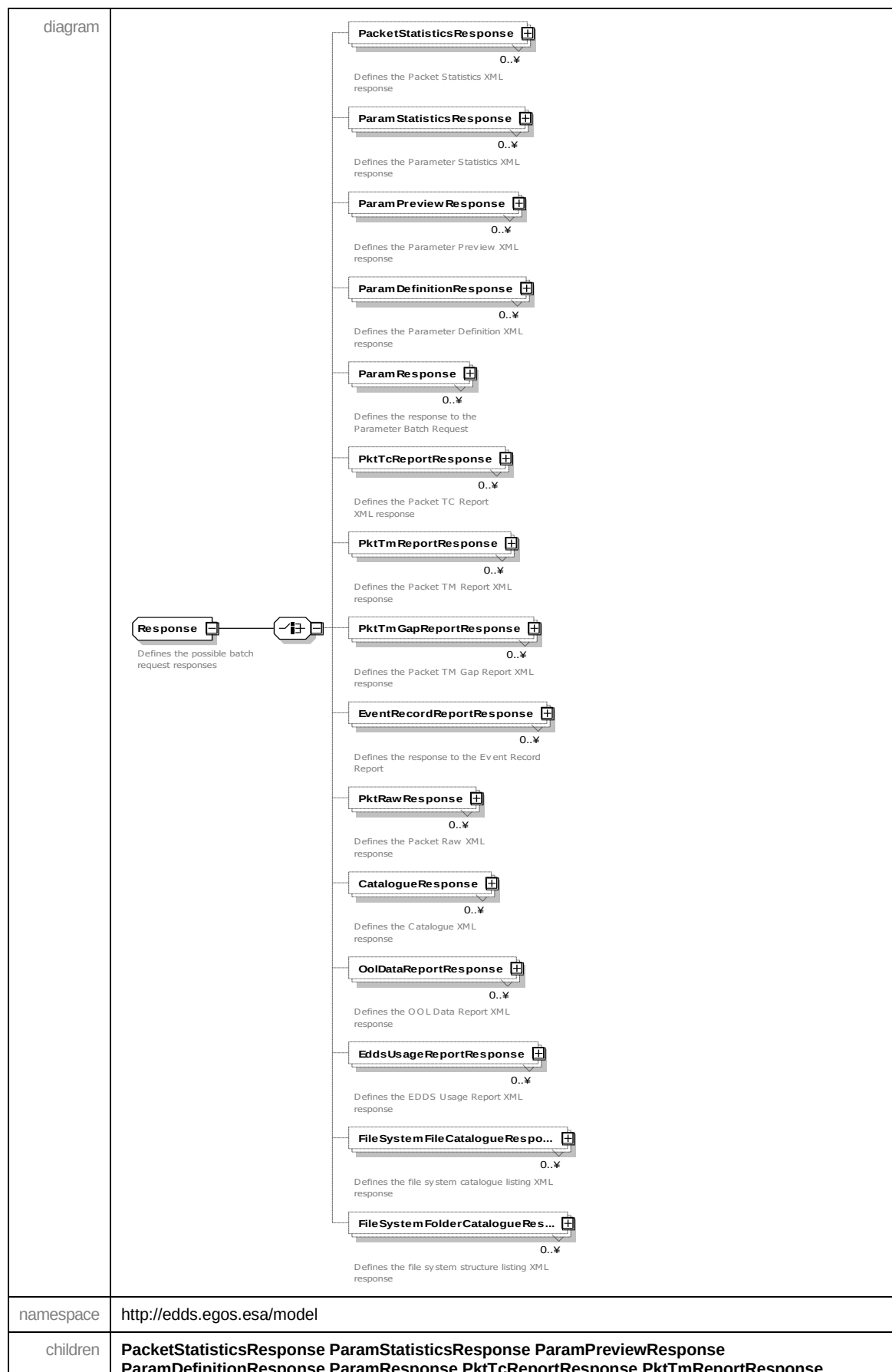
C.65 complexType PktTmStatistics

diagram	PktTmStatistics Defines the Telemetry Packet Statistics data request DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. TmPacketFilter Defines the basic information necessary to retrieve the data from the PARC archive and defines the filter to be applied to the retrieved data. PacketStatisticsFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	RequestPostProcessing Delivery TmPacketFilter PacketStatisticsFormat
used by	elements PktTmStatistics DataRequest/PktTmStatistics
annotation	documentation Defines the Telemetry Packet Statistics data request

C.66 complexType RequestMessagePart

diagram	RequestMessagePart Defines the message the client submit to server ContextPart Defines the context of the request. BatchRequest Defines the batch request.
namespace	http://edds.egos.esa/model
children	ContextPart BatchRequest
used by	element RequestMessagePart
annotation	documentation Defines the message the client submit to server

C.67 complexType Response



	PktTmGapReportResponse EventRecordReportResponse PktRawResponse CatalogueResponse OolDataReportResponse EddsUsageReportResponse FileSystemFileCatalogueResponse FileSystemFolderCatalogueResponse
used by	element ResponsePart/Response
annotation	documentation Defines the possible batch request responses

C.68 complexType ResponseData

diagram	The response files of a request The response files of a request 1..∞
namespace	http://edds.egos.esa/model
children	ResponseDataElement
used by	element ResponseData
annotation	documentation The response files of a request

C.69 complexType ResponseDataElement

diagram	The response files of a request The name of the original file The response file
namespace	http://edds.egos.esa/model
children	FileName Response
used by	element ResponseData/ResponseDataElement
annotation	documentation The response files of a request

C.70 complexType ResponsePart

diagram	Defines the XML response returned on user request Defines the response details
namespace	http://edds.egos.esa/model
children	Response
used by	element ResponsePart
annotation	documentation Defines the XML response returned on user request

C.71 complexType ResumePart

diagram	
namespace	http://edds.egos.esa/model
type	extension of InterruptPart
properties	base InterruptPart
children	User JobIdPart
used by	elements ResumePart ResumePartList/ResumePart
annotation	documentation Defines the message the client submits to resume a suspended or failed request

C.72 complexType ResumePartList

diagram	
namespace	http://edds.egos.esa/model
children	ResumePart
used by	element ResumePartList
annotation	documentation Defines a list of resume requests

C.73 complexType SMONParam

diagram	SMONParam (base type) Defines the Parameter data request for SMON DataRequestBase (extension) DataSource (optional) Defines the data source associated with the request RequestPostProcessing (optional) Defines if the request response has to be compressed or and encrypted. Delivery (optional) Defines the mechanism used to deliver the data response to the user. SMONParamFilter (optional) Defines the basic information necessary to retrieve the Parameter data from the DARC archive and defines the filter to be applied to the retrieved data. SMONParamFormat (optional) Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery SMONParamFilter SMONParamFormat
used by	elements SMONParam DataRequest/SmonParam
annotation	documentation Defines the Parameter data request for SMON

C.74 complexType SMONParamResponse

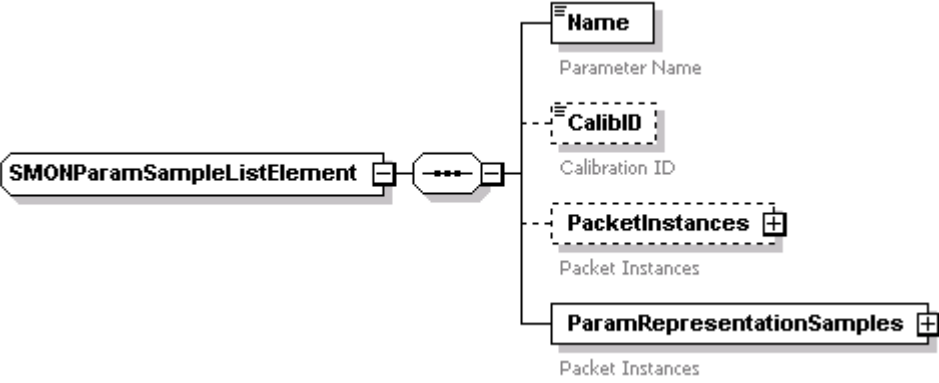
diagram	SMONParamResponse (base type) Defines the XML response returned on Parameter Batch Requests SMONParamSampleList (optional) Defines list of parameter samples
namespace	http://edds.egos.esa/model
children	SMONParamSampleList
annotation	documentation Defines the XML response returned on Parameter Batch Requests

C.75 complexType SMONParamResponseList

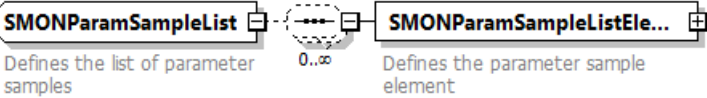
diagram	SMONParamSampleList (base type) Archive file filter descriptor SMONParamSampleListElement (optional) Defines the parameter sample element 0..∞
namespace	http://edds.egos.esa/model
children	SMONParamSampleListElement

used by	element SMONParamResponse/SMONParamSampleList
annotation	documentation Defines the list of parameter samples

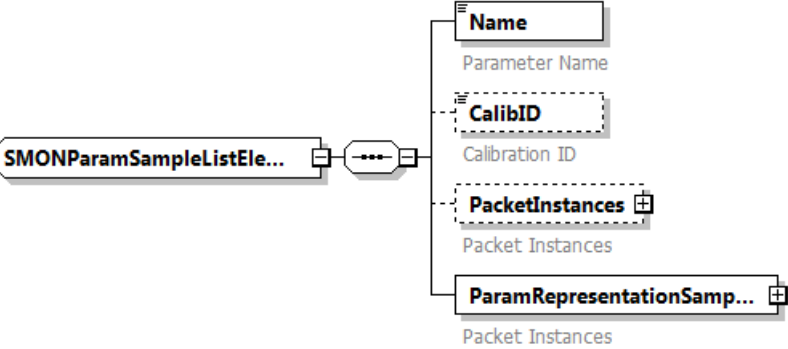
C.76 complexType SMONParamResponseListElement

diagram	 The diagram shows the structure of the SMONParamSampleListElement complex type. It consists of a root element SMONParamSampleListElement (solid box) connected to a sequence container (octagon with three dots). This container has four children: Name (solid box, Parameter Name), CalibID (dashed box, Calibration ID), PacketInstances (dashed box with a plus icon, Packet Instances), and ParamRepresentationSamples (solid box with a plus icon, Packet Instances).
namespace	http://edds.egos.esa/model
children	Name CalibID PacketInstances ParamRepresentationSamples
used by	elements SMONParamSampleListElement SMONParamSampleList/SMONParamSampleListElement

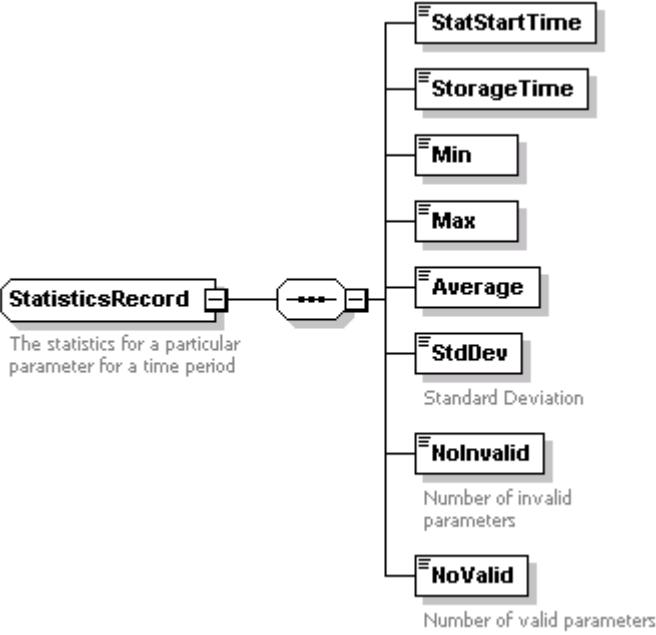
C.77 complexType SMONParamSampleList

diagram	 The diagram shows the structure of the SMONParamSampleList complex type. It consists of a root element SMONParamSampleList (solid box, Defines the list of parameter samples) connected to a sequence container (octagon with three dots, 0..∞) which contains one child element SMONParamSampleListEle... (solid box with a plus icon, Defines the parameter sample element).
namespace	http://edds.egos.esa/model
children	SMONParamSampleListElement
used by	elements SMONParamResponse

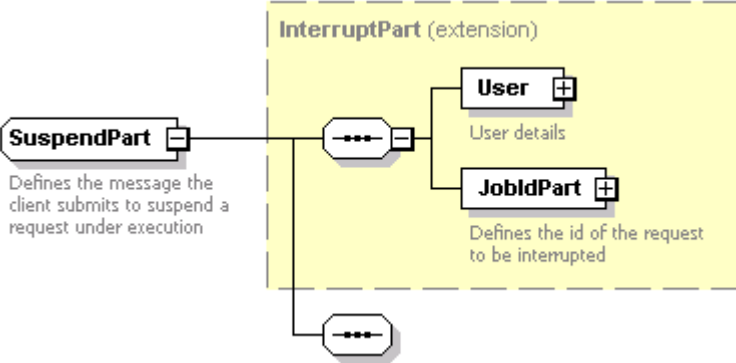
C.78 complexType SMONParamSampleListElement

diagram	 The diagram shows the structure of the SMONParamSampleListElement complex type. It consists of a root element SMONParamSampleListEle... (solid box) connected to a sequence container (octagon with three dots). This container has four children: Name (solid box, Parameter Name), CalibID (dashed box, Calibration ID), PacketInstances (dashed box with a plus icon, Packet Instances), and ParamRepresentationSamp... (solid box with a plus icon, Packet Instances).
namespace	http://edds.egos.esa/model
children	Name CalibID PacketInstances ParamRepresentationSamples
used by	elements SMONParamSampleList

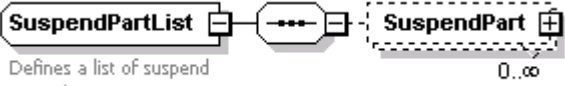
C.79 complexType StatisticsRecord

diagram	 The statistics for a particular parameter for a time period StatStartTime StorageTime Min Max Average StdDev Standard Deviation NolInvalid Number of invalid parameters NoValid Number of valid parameters
namespace	http://edds.egos.esa/model
children	StatStartTime StorageTime Min Max Average StdDev NolInvalid NoValid
used by	element ParamStatisticListElement/StatisticsRecord
annotation	documentation The statistics for a particular parameter for a time period

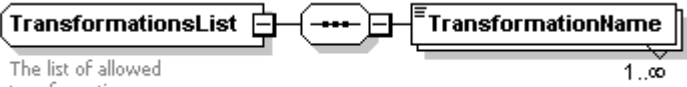
C.80 complexType SuspendPart

diagram	 SuspendPart Defines the message the client submits to suspend a request under execution InterruptPart (extension) User User details JobIdPart Defines the id of the request to be interrupted
namespace	http://edds.egos.esa/model
type	extension of InterruptPart
properties	base InterruptPart
children	User JobIdPart
used by	elements SuspendPart SuspendPartList/SuspendPart
annotation	documentation Defines the message the client submits to suspend a request under execution

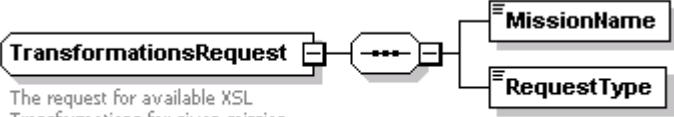
C.81 complexType SuspendPartList

diagram	 Defines a list of suspend requests
namespace	http://edds.egos.esa/model
children	SuspendPart
used by	element SuspendPartList
annotation	documentation Defines a list of suspend requests

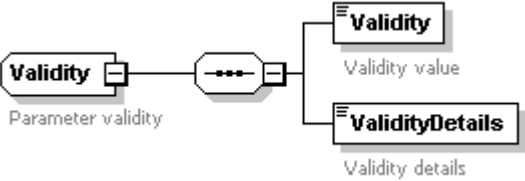
C.82 complexType TransformationsList

diagram	 The list of allowed transformation names
namespace	http://edds.egos.esa/model
children	TransformationName
used by	element TransformationsList
annotation	documentation The list of allowed transformation names

C.83 complexType TransformationsRequest

diagram	 The request for available XSL Transformations for given mission and request type
namespace	http://edds.egos.esa/model
children	MissionName RequestType
used by	element TransformationsRequest
annotation	documentation The request for available XSL Transformations for given mission and request type

C.84 complexType Validity

diagram	 Parameter validity
namespace	http://edds.egos.esa/model
children	Validity ValidityDetails
used by	element ParamRepresentationSample/Validity

annotation	documentation Parameter validity
------------	-------------------------------------

C.85 simpleType AliasInternalType

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	
	enumeration	DATAPROVIDER	
	enumeration	BATCH	
	enumeration	STREAM	

C.86 simpleType DeltaBehaviourState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	
	enumeration	DELTA_ALLOWED	
	enumeration	DELTA_EQUAL	
	enumeration	MAX_DELTA_LOW	
	enumeration	MAX_DELTA_HIGH	
	enumeration	MIN_DELTA_LOW	
	enumeration	MIN_DELTA_HIGH	

C.87 simpleType Direction

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	
	enumeration	ASC	
	enumeration	DESC	

C.88 simpleType FilterType

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	AND	documentation The different filter options should all be satisfied.
	enumeration	OR	documentation One of the filter options should be satisfied.
annotation	documentation Whether the filters should be applied with an AND or OR condition.		

C.89 simpleType LimitBehaviourState

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	WITHIN_RANGE
	enumeration	LOW
	enumeration	HIGH
	enumeration	LOW_LOW
	enumeration	HIGH_HIGH

C.90 simpleType SCCBehaviourState

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	SCC_ENABLE
	enumeration	SCC_UNINIT
	enumeration	SCC_DISABLE
	enumeration	SCC_OFF
	enumeration	SCC_FAIL

C.91 simpleType StatusBehaviourState

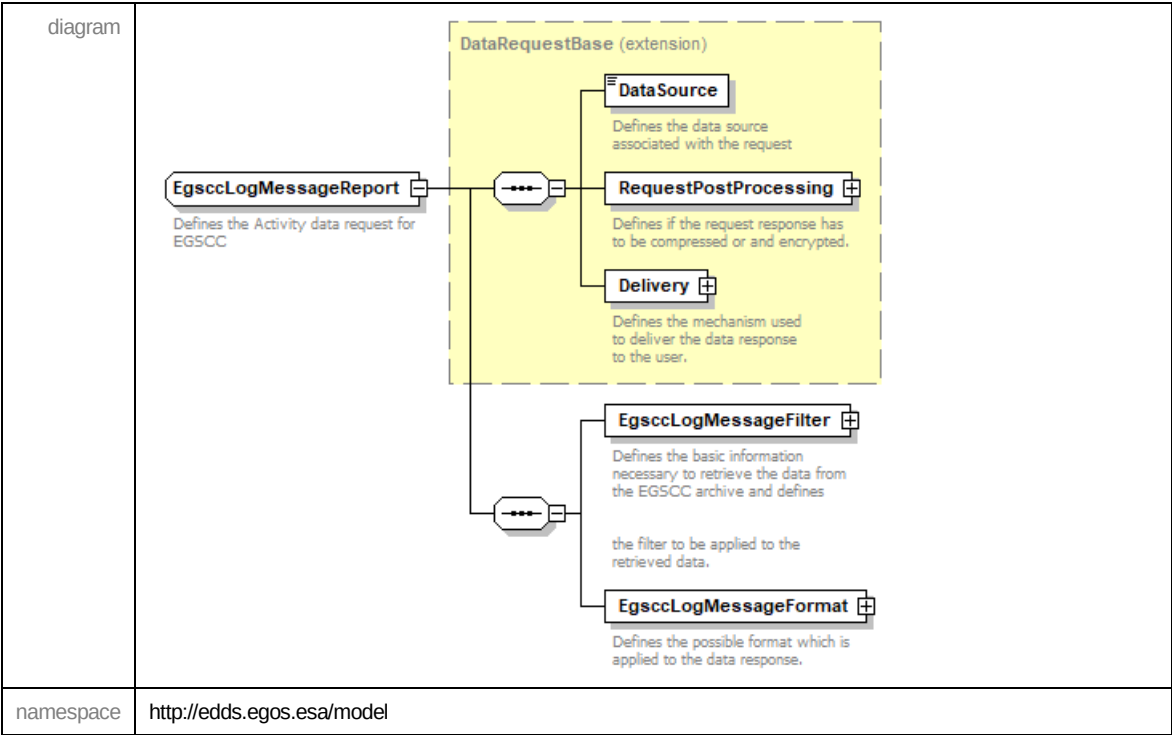
namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	ALLOWED
	enumeration	UNEXPECTED

C.92 simpleType ValidityValue

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	VAL_VALID
	enumeration	VAL_STATE_OFF
	enumeration	VAL_POWER_OFF
	enumeration	VAL_ROUTE_OFF
	enumeration	VAL_PRESERVE
	enumeration	VAL_TRANSIENT
	enumeration	VAL_TCO_ERROR
	enumeration	VAL_UNUSED_1

	enumeration	VAL_UNUSED_2
	enumeration	VAL_UNUSED_3
	enumeration	VAL_UNUSED_4
	enumeration	VAL_UNUSED_5
	enumeration	VAL_UNUSED_6
	enumeration	VAL_UNUSED_7
	enumeration	VAL_EXPIRED
	enumeration	VAL_UNKNOWN_STATE
	enumeration	VAL_UNKNOWN_CONDITION
	enumeration	VAL_UNKNOWN_CRITERIA
	enumeration	VAL_CALIBRATION
	enumeration	VAL_TOO_EARLY
	enumeration	VAL_UNKNOWN_PKT
	enumeration	VAL_UNINIT
	enumeration	VAL_PKT_RETRV
	enumeration	VAL_MIB_ERROR
	enumeration	VAL_SYSTEM_ERROR
	enumeration	VAL_FIELD_UNKNOWN
	enumeration	VAL_FIELD_ABSENT
	enumeration	VAL_UNKNOWN_TYPE
	enumeration	VAL_UNKNOWN_OP
	enumeration	VAL_UNKNOWN_CONV
	enumeration	VAL_OVERFLOW
	enumeration	VAL_DIVIDE_BY_ZERO
	enumeration	VAL_OL_PARSE

C.93 complexType EgscclogMessageReport



children	DataSource RequestPostProcessing Delivery EgscclLogMessageFilter EgscclLogMessageFormat
used by	element

C.94 complexType EgscclRestLogMessageReport

diagram	EgscclRestLogMessageReport Defines the Log Message REST request for EGSCC DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing + Defines if the request response has to be compressed or and encrypted. Delivery + Defines the mechanism used to deliver the data response to the user. EgscclRestLogMessageFilter + Defines the basic information necessary to retrieve the data from the EGSCC REST API and defines the filter to be applied to the retrieved data. EgscclRestLogMessageFormat + Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgscclRestLogMessageFilter EgscclRestLogMessageFormat
used by	element

C.95 complexType EgsccRestPktTmReport

diagram	
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgsccRestTmPacketFilter ReportFormat
used by	element

C.96 complexType EgsccRestPktTcReport

diagram	EgsccRestPktTcReport Defines filter to be applied for the TC report DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. EgsccRestTcPacketFilter Defines the basic information necessary to retrieve the data from the Report and defines the filter to be applied to the retrieved data. ReportFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgsccRestTcPacketFilter ReportFormat
used by	element

C.97 complexType EgsccRestParam

diagram	EgsccRestParam Defines the Parameter data request DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. EgsccRestParamTmFilter Defines the basic information necessary to retrieve the Parameter data using EGSCC REST API and defines the filter to be applied to the retrieved data. EgsccRestParamFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgsccRestParamTmFilter EgsccRestParamFormat
used by	element

C.98 complexType EgscRestPktTm

diagram	The diagram illustrates the structure of the EgscRestPktTm complexType. It is composed of several elements: EgscRestPktTm: Defines the Telemetry Packet data request for EGSCC rest request. DataRequestBase (extension): A dashed box containing three elements: DataSource: Defines the data source associated with the request. RequestPostProcessing: Defines if the request response has to be compressed or and encrypted. Delivery: Defines the mechanism used to deliver the data response to the user. EgscRestTmPacketFilter: Defines the basic information necessary to retrieve the data from the EGSCC rest service and defines the filter to be applied to the retrieved data. PacketFormat: Defines the possible format which is applied to the data response. Connections: EgscRestPktTm is connected to DataRequestBase and EgscRestTmPacketFilter. DataRequestBase is connected to RequestPostProcessing and Delivery. EgscRestTmPacketFilter is connected to PacketFormat.
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgscRestTmPacketFilter PacketFormat
used by	element

C.99 complexType EgsccRestActivityReport

diagram	
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgsccRestActivityFilter EgsccActivityFormat
used by	element

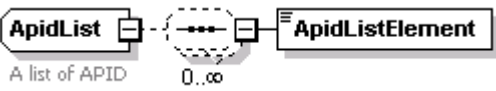
C.100 complexType EgsccRestOolReport

diagram	
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgsccRestOolFilter ReportFormat
used by	element

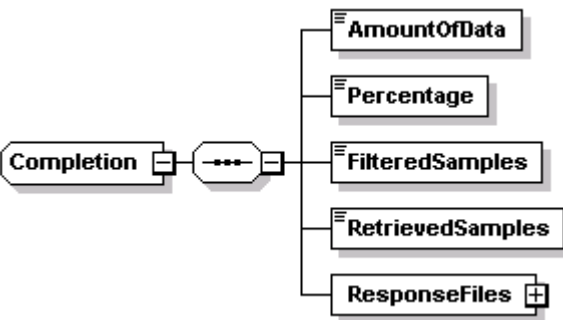
Appendix D Common Data Types

This section of the document describes the common complex types used for all services as defined in the common.xsd file.

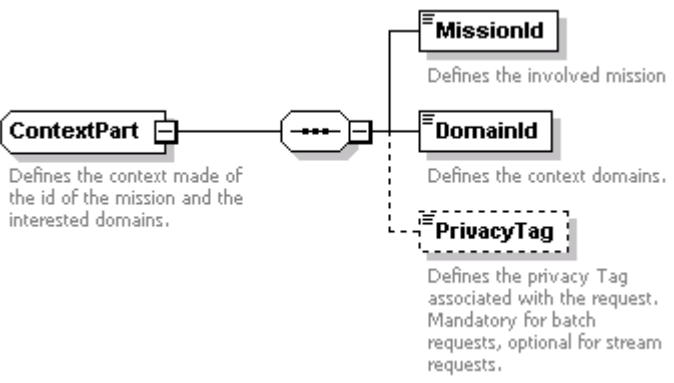
D.1 *complexType ApidList*

diagram	 A list of APID 0..∞
namespace	http://edds.egos.esa/model
children	ApidListElement
annotation	documentation A list of APID

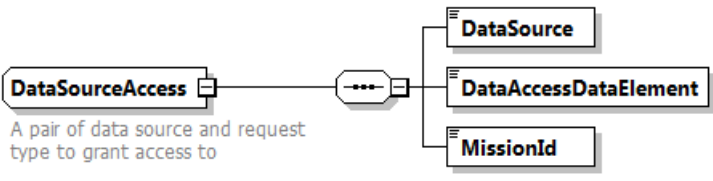
D.2 *complexType Completion*

diagram	
namespace	http://edds.egos.esa/model
children	AmountOfData Percentage FilteredSamples RetrievedSamples ResponseFiles

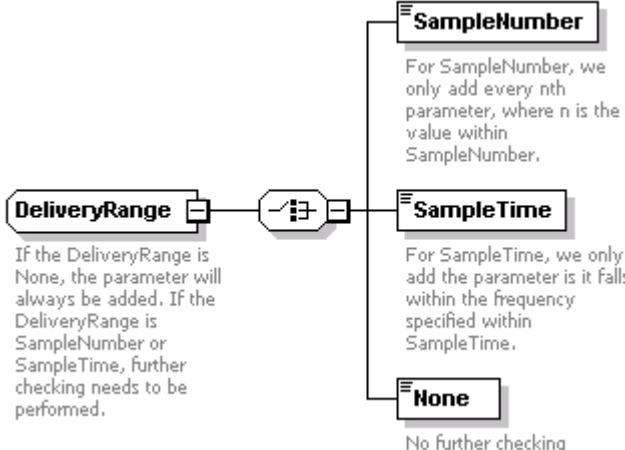
D.3 *complexType ContextPart*

diagram	 ContextPart Defines the context made of the id of the mission and the interested domains. MissionId Defines the involved mission DomainId Defines the context domains. PrivacyTag Defines the privacy Tag associated with the request. Mandatory for batch requests, optional for stream requests.
namespace	http://edds.egos.esa/model
children	MissionId DomainId PrivacyTag
annotation	documentation Defines the context made of the id of the mission and the interested domains.

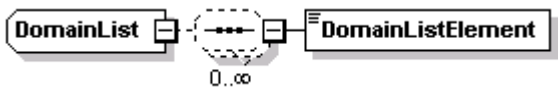
D.4 complexType DataSourceAccess

diagram	 A pair of data source and request type to grant access to
namespace	http://edds.egos.esa/model
children	DataSource DataAccessDataElement string
annotation	documentation A pair of data source and request type to grant access to

D.5 complexType DeliveryRange

diagram	 If the DeliveryRange is None, the parameter will always be added. If the DeliveryRange is SampleNumber or SampleTime, further checking needs to be performed. For SampleNumber, we only add every nth parameter, where n is the value within SampleNumber. For SampleTime, we only add the parameter is it falls within the frequency specified within SampleTime. No further checking
namespace	http://edds.egos.esa/model
children	SampleNumber SampleTime None
annotation	documentation If the DeliveryRange is None, the parameter will always be added. If the DeliveryRange is SampleNumber or SampleTime, further checking needs to be performed.

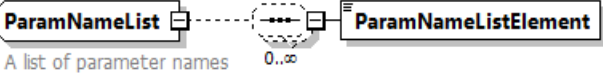
D.6 complexType DomainList

diagram	 0..∞
namespace	http://edds.egos.esa/model
children	DomainListElement

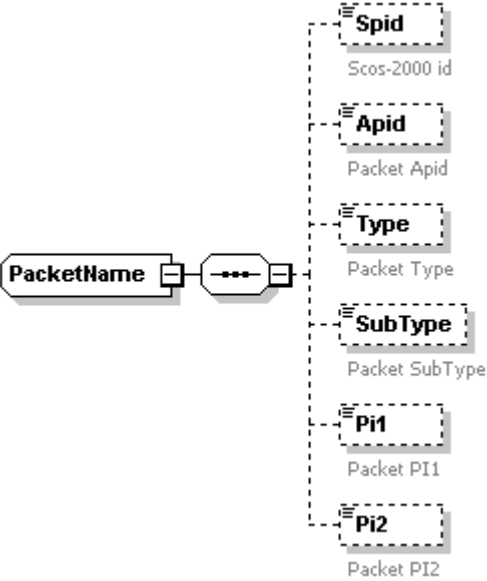
D.7 complexType EmailList

diagram	 Defines multiple e-mail address 0..∞ Single E-mail address
namespace	http://edds.egos.esa/model
children	EmailListElement
annotation	documentation Defines multiple e-mail address

D.8 complexType ParamNameList

diagram	 A list of parameter names 0..∞
namespace	http://edds.egos.esa/model
children	ParamNameListElement
annotation	documentation A list of parameter names

D.9 complexType PacketName

diagram	
namespace	http://edds.egos.esa/model
children	Spid Apid Type SubType Pi1 Pi2
used by	element PacketNameList/PackageNameListElement

D.10 complexType PacketNameList

diagram	PacketNameList List of generated files 0..∞ PacketNameListElement List of packets to be retrieved. Note that each element of the packet name supports wildcards * and ? and negation ~
namespace	http://edds.egos.esa/model
children	PacketNameListElement
annotation	documentation List of generated files

complexType ParamNameList

diagram	ParamNameList A list of parameter names 0..∞ ParamNameListElement
namespace	http://edds.egos.esa/model
children	ParamNameListElement
annotation	documentation A list of parameter names

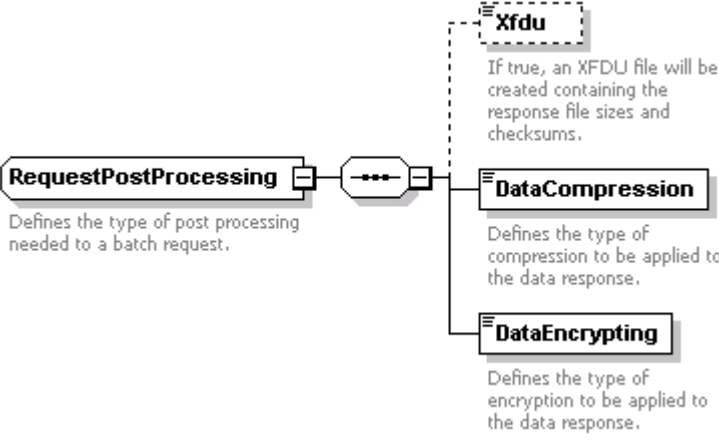
D.11 complexType Pi1List

diagram	Pi1List 0..∞ Pi1ListElement
namespace	http://edds.egos.esa/model
children	Pi1ListElement

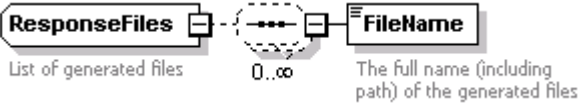
D.12 complexType Pi2List

diagram	Pi2List 0..∞ Pi2ListElement
namespace	http://edds.egos.esa/model
children	Pi2ListElement

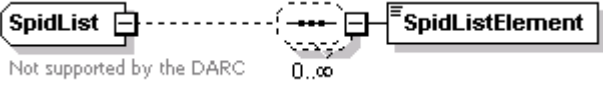
D.13 complexType RequestPostProcessing

diagram	
namespace	http://edds.egos.esa/model
children	DataCompression DataEncrypting
annotation	documentation Defines the type of post processing needed to a batch request.

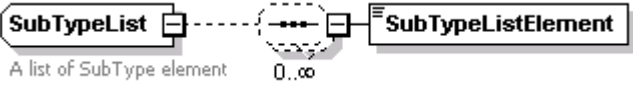
D.14 complexType ResponseFiles

diagram	
namespace	http://edds.egos.esa/model
children	FileName
used by	element Completion/ResponseFiles
annotation	documentation List of generated files

D.15 complexType SpidList

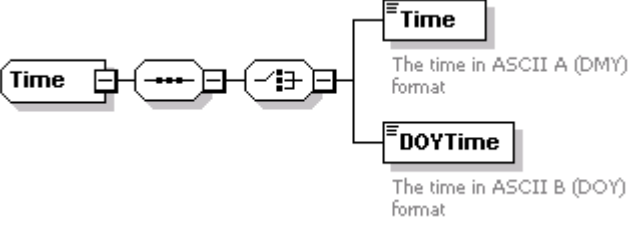
diagram	
namespace	http://edds.egos.esa/model
children	SpidListElement
annotation	documentation Not supported by the DARC

D.16 complexType SubTypeList

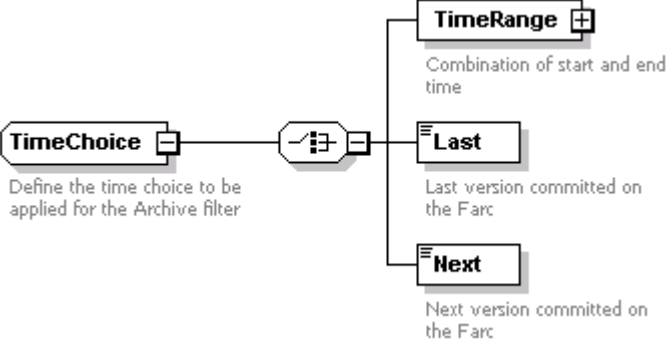
diagram	
---------	--

namespace	http://edds.egos.esa/model
children	SubTypeListElement
annotation	documentation A list of SubType element

D.17 complexType Time

diagram	
namespace	http://edds.egos.esa/model
children	Time DOYTime

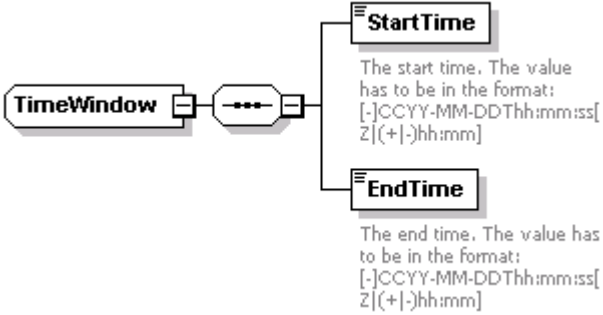
D.18 complexType TimeChoice

diagram	
namespace	http://edds.egos.esa/model
children	TimeRange Last Next
annotation	documentation Define the time choice to be applied for the Archive filter

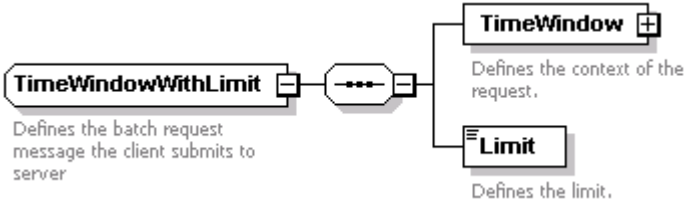
D.19 complexType TimeRange

diagram	
namespace	http://edds.egos.esa/model
children	StartTime StartDOYTime RelativeStartTime EndTime EndDOYTime Duration ConsolidationTime
used by	element EddsUsageReportFilter/TimeRange EvPacketFilter/TimeRange ParamPreviewFilter/TimeRange ParamStatisticsFilter/TimeRange ParamTmFilter/TimeRange OolDataReportFilter/TimeRange TcPacketFilter/TimeRange TmPacketFilter/TimeRange RawPacketFilter/TimeRange TimeChoice/TimeRange

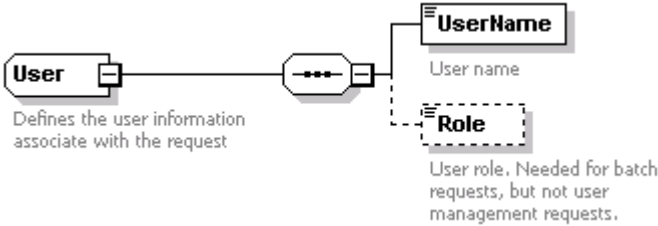
D.20 complexType TimeWindow

diagram	
namespace	http://edds.egos.esa/model
children	StartTime EndTime
used by	element TimeWindow

D.21 complexType TimeWindowWithLimit

diagram	
namespace	http://edds.egos.esa/model
children	TimeWindow Limit
used by	element TimeWindowWithLimit
annotation	documentation Defines the batch request message the client submits to server

D.22 complexType User

diagram	
namespace	http://edds.egos.esa/model
children	UserName Role
annotation	documentation Defines the user information associate with the request

D.23 simpleType AccountRequestSubType

namespace	http://edds.egos.esa/model
-----------	----------------------------

type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	Group	
	enumeration	Role	
	enumeration	OperationSet	
	enumeration	QuotaSet	
	enumeration	DataAccessSet	
	enumeration	Mission	
	enumeration	UserAccount	

D.24 simpleType DataAccessDataElement

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	Param	
	enumeration	ParamStatistics	
	enumeration	ParamDefinition	
	enumeration	ParamPreview	
	enumeration	SmonParam	
	enumeration	PktEv	
	enumeration	PktTm	
	enumeration	PktTc	
	enumeration	PktTmStatistics	
	enumeration	PktTcStatistics	
	enumeration	PktEvStatistics	
	enumeration	PktEvRaw	
	enumeration	PktTmRaw	
	enumeration	PktTcRaw	
	enumeration	PktTmReport	
	enumeration	PktTcReport	
	enumeration	PktTmGapReport	
	enumeration	EventRecordReport	
	enumeration	ArchiveCatalogue	
	enumeration	ArchiveFile	
	enumeration	ArchiveSubscription	
	enumeration	OolRecordReport	
	enumeration	EddsUsageReport	
	enumeration	ParamStream	
	enumeration	PktEvStream	
	enumeration	PktTmStream	
	enumeration	PktTcStream	
	enumeration	OolStream	
	enumeration	FileSystemFileCatalogue	
	enumeration	FileSystemFolderCatalogue	
	enumeration	FileSystemFile	
	enumeration	FileSystemSubscription	
	enumeration	EgscclActivityReport	

	enumeration	EgscclLogMessageReport
--	-------------	------------------------

D.25 simpleType DataCompression

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	RequestPostProcessing/DataCompression	
facets	Kind	Value	annotation
	enumeration	ZIP	
	enumeration	TAR	
	enumeration	TARGZ	
	enumeration	NONE	
annotation	documentation	Defines the possible compression algorithm which can be applied to the data responses.	

D.26 simpleType DataEncrypting

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	RequestPostProcessing/DataEncrypting	
facets	Kind	Value	annotation
	enumeration	NONE	
	enumeration	AES	
annotation	documentation	Defines the data encryption to be applied to the data responses. The implementation of the encryption algorithm is mission specific.	

D.27 simpleType DataSource

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	FARC	
	enumeration	DARC	
	enumeration	PARC	
	enumeration	EDDS	
	enumeration	SMON	
	enumeration	FileSystem	
	enumeration	DataProvisionService	
annotation	documentation	Defines data source accociated with the request	

D.28 simpleType Direction

namespace	http://edds.egos.esa/model		
-----------	----------------------------	--	--

type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	ASC	
	enumeration	DESC	
annotation	documentation Defines whether to retrieve ascending or descending. If not set, the default is descending.		

D.29 simpleType DOYDateTime

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	elements	EddsUsageReportFilterKeyword/CompletionDOYTime Time/DOYTime TimeRange/EndDOYTime EddsUsageReportFilterKeyword/ExecutionDOYTime PktTcFilterKeyword/ExecutionDOYTime PktTmFilterKeyword/ExtractionDOYTime PktTmStreamFilterKeyword/ExtractionDOYTime PktEvFilterKeyword/GenerationDOYTime PktTmFilterKeyword/GenerationDOYTime ParamFilterKeyword/OnBoardDOYTime PktTcFilterKeyword/ReleaseDOYTime TimeRange/StartDOYTime EddsUsageReportFilterKeyword/SubmissionDOYTime	
facets	Kind pattern	Value \\d{4}-(00[1-9] 0[1-9]\\d [12]\\d{2})3([0-5]\\d 6[0-6]))T([01]\\d 2[0-3]):[0-5]\\d:[0-5]\\d(\\.\\d{1,6})?(Z)?	annotation
annotation	documentation Defines the YYYY-DDD'T'HH:MM:SS[.SSSSSSZ] format with rules for days: 001-366, hours:00-23, minutes: 00-59, seconds: 00-59, optional fraction of seconds .0-.999999 and optional Zulu time indicator		

D.30 simpleType NamePrefixSuffixElement

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	default	
	enumeration	filename	
	enumeration	prefix	
	enumeration	suffix	
annotation	documentation Defines the possible ways to use the file text.		

D.31 simpleType BehaviourState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	Param	
	enumeration	LIMIT_HIGH	
	enumeration	LIMIT_HIGH_HIGH	
	enumeration	LIMIT_LOW	

	enumeration	LIMIT_LOW_LOW
	enumeration	LIMIT_WITHIN_RANGE

D.32 simpleType OolBehaviourDeltaState

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	LIMIT_HIGH
	enumeration	LIMIT_HIGH_HIGH
	enumeration	OUT_OF_LIMIT_LOW
	enumeration	LIMIT_LOW_LOW
	enumeration	LIMIT_WITHIN_RANGE

D.33 simpleType OolBehaviourLimitState

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	LIMIT_HIGH
	enumeration	LIMIT_HIGH_HIGH
	enumeration	LIMIT_LOW
	enumeration	LIMIT_LOW_LOW
	enumeration	LIMIT_WITHIN_RANGE

D.34 simpleType OolBehaviourSccState

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	SCC_ENABLE
	enumeration	SCC_DISABLE
	enumeration	SCC_FAIL
	enumeration	SCC_OFF
	enumeration	SCC_UNINIT

D.35 simpleType OolBehaviourStatusState

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
facets	Kind	Value
	enumeration	STATUS_ALLOWED
	enumeration	STATUS_UNEXPECTED

D.36 simpleType OolParameterState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind enumeration	Value NOMINAL	annotation
	enumeration	WARNING	
	enumeration	OUT_OF_LIMITS	
	enumeration	SCC	
	enumeration	SCC_DISABLE	
	enumeration	SCC_OFF	
	enumeration	SCC_UNINIT	
	enumeration	VIOLATION	
	enumeration	NO_VALUE	documentation NO_VALUE is used if a value is returned from the archive that doesn't match a value from this enumerated list

D.37 simpleType OolRecordType

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind enumeration	Value LIMIT	annotation
	enumeration	STATE	
	enumeration	STATUS_CONSISTENCY	
	enumeration	SOFT_HARD_LIMIT	
	enumeration	LIMIT_MO	
	enumeration	DELTA	
	enumeration	NO_VALUE	documentation NO_VALUE is used if a value is returned from the archive that doesn't match a value from this enumerated list

D.38 simpleType OolState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind enumeration	Value INCONSISTENT_STATUS	annotation
	enumeration	NOMINAL	
	enumeration	OUT_OF_LIMIT	
	enumeration	UNCHECKED	
	enumeration	UNDEFINED	
	enumeration	VIOLATED	
	enumeration	WARNING	

D.39 simpleType Operation

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	OP_EQ	
	enumeration	OP_NEQ	
	enumeration	OP_GT	
	enumeration	OP_GTE	
	enumeration	OP_LT	
	enumeration	OP_LTE	
	enumeration	OP_NOT	
	enumeration	OP_MATCH	
	enumeration	OP_CONTAINS	documentation Used for container objects; matches lists containing the specified filter item, but the container is allowed to contain other items not specified.

D.40 simpleType OrderBy

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	creation_time ASC	
	enumeration	creation_time DESC	
	enumeration	scheduled_execution_time ASC	
	enumeration	scheduled_execution_time DESC	
	enumeration	username ASC	
	enumeration	username DESC	
annotation	documentation Defines the possible compression algorithm which can be applied to the data responses.		

D.41 simpleType ParamValidityStatus

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	VALID	
	enumeration	INVALID	
	enumeration	UNKNOWN	
	enumeration	EXPIRED	
annotation	documentation The possible validity states of a parameter		

D.42 simpleType PrivacyTag

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ContextPart/PrivacyTag	
facets	Kind enumeration	Value PRIVATE	annotation
	enumeration	ROLE	
	enumeration	MISSION	
	enumeration	PUBLIC	
annotation	documentation Defines privacy tag values which can be associated to a request		

D.43 simpleType SpidsSpecifier

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	elements	PacketName/Apid PacketName/Pi1 PacketName/Pi2 PacketName/Spid PacketName/SubType PacketName/Type	
facets	Kind pattern	Value (\s*((\d+\s*\s*\d+))[\^,\s]+)\s*(,\s*((\d+\s*\s*\d+))[\^,\s]+)\s*)*)\s*	annotation
annotation	documentation A string with regex restriction to allow for a comma separated list of both string values and numeric value ranges.		

D.44 simpleType State

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind enumeration	Value SUBMITTED	annotation documentation The request reached the web server and has been saved into the DB
	enumeration	CANCELED	documentation The request has been cancelled
	enumeration	SUSPENDED	documentation The request has been suspended
	enumeration	QUEUED	documentation A submitted request is taken from the DB and queued for execution in the edds-server
	enumeration	COMPLETED_NO_RESULTS	documentation The edds-server completed the elaboration of a request and no results were received from the archive
	enumeration	SERVER_COMPLETED	documentation The edds-server completed the elaboration of a request and dispatch the result to the delivery-manager
	enumeration	DELIVERED	documentation The request execution is completed
	enumeration	DELIVERED_RESP_DELETED	documentation The request execution is completed and the data on EDDS file server is deleted
	enumeration	DELIVERED_PARTIAL_RESULTS	documentation The request execution is completed, but not all of the requested data could be retrieved
	enumeration	ERROR_LOCALLY_DELIVERED	documentation The request execution is completed, but the

	enumerationERROR_ACCESS_DENIED enumerationERROR_INVALID_REQUEST enumerationERROR_LIMIT_EXCEEDED enumerationERROR_SERVER_NOT_AVAILABLE enumerationERROR_DELIVERY enumerationERROR_UNKNOWN	delivery manager was not able to dispatch the result through FTP. The file is available in the EDDS Server FTP directory documentation The user does not have the right privileges to execute the request documentation The request is not valid documentation The quota associated with the user has been exceeded documentation The needed data provider is not available documentation The delivery manager was not able to deliver the file documentation Not predictable error in the request processing
annotation	documentation Request execution statuses	

D.45 simpleType MessageSeverity

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
facets	Kind	Value	annotation
	enumeration	ALARM	
	enumeration	WARNING	
	enumeration	INFO	
	enumeration	DEBUG	
annotation	documentation The possible severity states of a log message		

D.46 simpleType MessageType

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
facets	Kind	Value	annotation
	enumeration	LOG	
	enumeration	OPERATIONAL	
	enumeration	USER	
	enumeration	TRACE	
annotation	documentation The possible message type states of a log message		

D.47 simpleType MessageAcknowledgementType

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	POSITIVE	
	enumeration	NEGATIVE	
annotation	documentation The possible message acknowledgement type of a log message		

D.48 simpleType AlarmState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	NOT_CHECKED	
	enumeration	NOMINAL	
	enumeration	WARNING	
	enumeration	ALARM	
	enumeration	NOMINAL_NO_CHECK_DEFINED	
	enumeration	NOMINAL_CHECKS_DISABLED	
	enumeration	UNKNOWN_STATE	
	enumeration	VIOLATED	
annotation	documentation The possible alarm state		

D.49 simpleType ProcessingState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	PROCESSED	
	enumeration	FORCED	
	enumeration	DISABLED	
	enumeration	SUSPENDED	
	enumeration	NOT_RECEIVED	
	enumeration	INVALID_DEFINITION	
	enumeration	ERROR	
annotation	documentation The possible processing state		

D.50 simpleType ValidityType

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	VALID	
	enumeration	UNKNOWN	
	enumeration	INVALID	
	enumeration	UNKNOWN_VALUE	
annotation	documentation The possible validity type		

D.51 simpleType ActivityState

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	NOT_INITIATED	
	enumeration	PRECONDITIONS	
	enumeration	PRECONDITIONS_SCHEDULED	
	enumeration	ROUTING	
	enumeration	ROUTING_SCHEDULED	
	enumeration	ROUTING_WAIT_EVENT	
	enumeration	ROUTING_RELEASED	
	enumeration	ROUTING_RADIATED	
	enumeration	ROUTING_RECEIVED	
	enumeration	EXECUTING	
	enumeration	EXECUTING_PAUSING	
	enumeration	EXECUTING_PAUSED	
	enumeration	EXECUTING_RESUMING	
	enumeration	COMPLETED_SUCCESS	
	enumeration	COMPLETED_FAILURE	
	enumeration	COMPLETED_TIMEOUT	
	enumeration	COMPLETED_ABORTED	
	enumeration	COMPLETED_STOPPED	
	enumeration	CONFIRMATION_PENDING	
	enumeration	CONFIRMATION_SUCCESS	
	enumeration	CONFIRMATION_TIMEOUT	
	enumeration	CONFIRMATION_FAILURE	
	enumeration	COMPLETED_UNVERIFIED	
	enumeration	CANCELLED_DELETED	
	enumeration	CANCELLED_DISABLED	
annotation	documentation The possible activity state.		

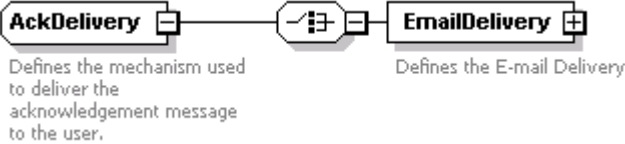
D.52 simpleType PriorityLevel

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	EXPEDITED	
	enumeration	STANDARD	
annotation	documentation The possible priority level		

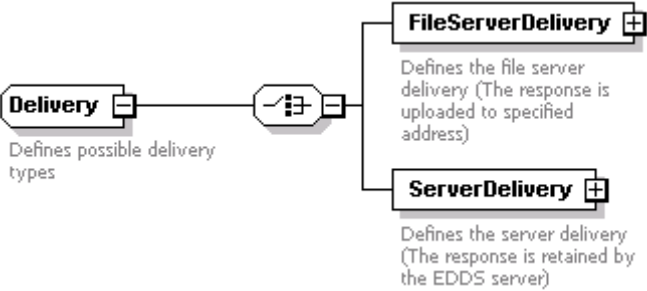
Appendix E Delivery Data Type

This section of the document describes the complex types used for delivery as defined in the delivery.xsd file.

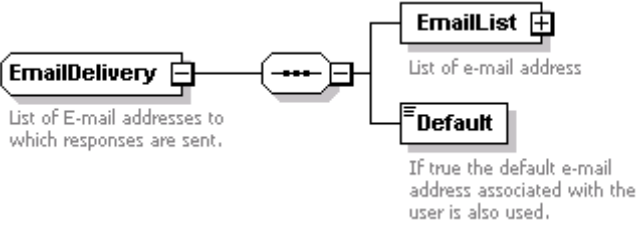
E.1 complexType AckDelivery

diagram	 Defines the mechanism used to deliver the acknowledgement message to the user. Defines the E-mail Delivery
namespace	http://edds.egos.esa/model
children	EmailDelivery
annotation	documentation Defines the mechanism used to deliver the acknowledgement message to the user.

E.2 complexType Delivery

diagram	 Defines possible delivery types Defines the file server delivery (The response is uploaded to specified address) Defines the server delivery (The response is retained by the EDDS server)
namespace	http://edds.egos.esa/model
children	FileServerDelivery ServerDelivery
annotation	documentation Defines possible delivery types

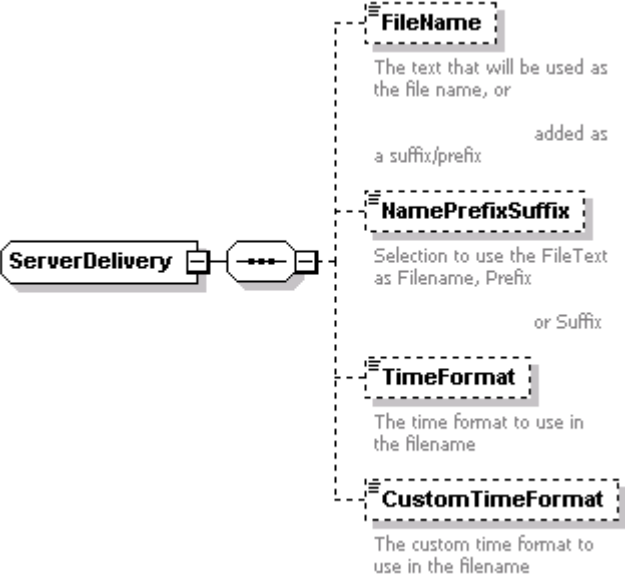
E.3 complexType EmailDelivery

diagram	 List of E-mail addresses to which responses are sent. List of e-mail address If true the default e-mail address associated with the user is also used.
namespace	http://edds.egos.esa/model
children	EmailList Default
used by	element AckDelivery/EmailDelivery
annotation	documentation List of E-mail addresses to which responses are sent.

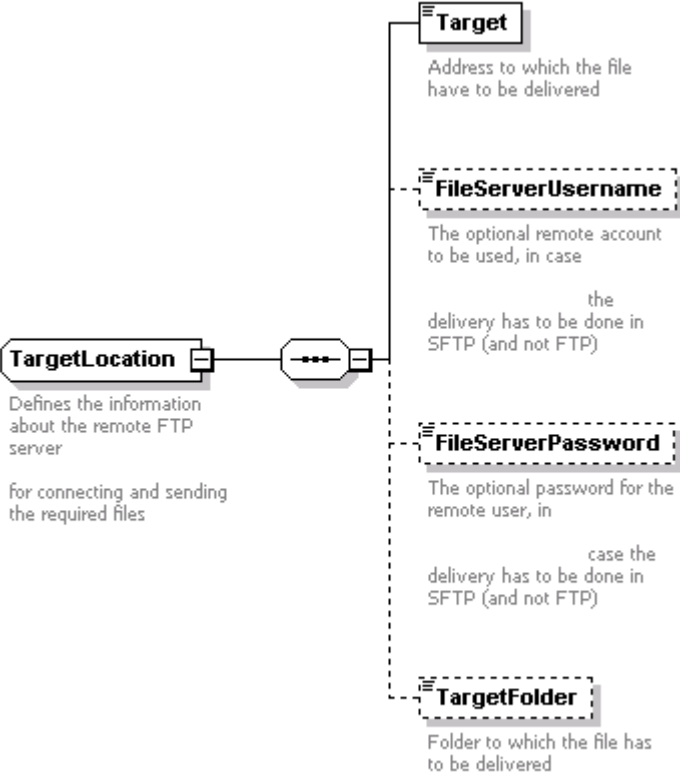
E.4 complexType FileServerDelivery

diagram	The diagram shows the FileServerDelivery complex type. It is connected to a sequence container (rectangle with three dots) which contains five elements: TargetLocations (1..∞), FileName, NamePrefixSuffix, KeepFileAfterDelivery, TimeFormat, and CustomTimeFormat. Each element has a descriptive text below it. TargetLocations 1..∞ FileName The text that will be used as the file name, or added as a suffix/prefix NamePrefixSuffix Selection to use the FileText as Filename, Prefix or Suffix KeepFileAfterDelivery Defines if the data responses has to be deleted after a successful delivery. By default the file is deleted from the EDDS server. TimeFormat The time format to use in the filename CustomTimeFormat The custom time format to use in the filename
namespace	http://edds.egos.esa/model
children	TargetLocations FileName NamePrefixSuffix KeepFileAfterDelivery TimeFormat CustomTimeFormat
used by	element Delivery/FileServerDelivery

E.5 complexType ServerDelivery

diagram	
namespace	http://edds.egos.esa/model
children	FileName NamePrefixSuffix TimeFormat CustomTimeFormat
used by	element Delivery/ServerDelivery

E.6 complexType TargetLocation

diagram	
namespace	http://edds.egos.esa/model
children	Target FileServerUsername FileServerPassword TargetFolder

used by	element FileServerDelivery/TargetLocations
annotation	documentation Defines the information about the remote FTP server for connecting and sending the required files

E.7 *simpleType FileName*

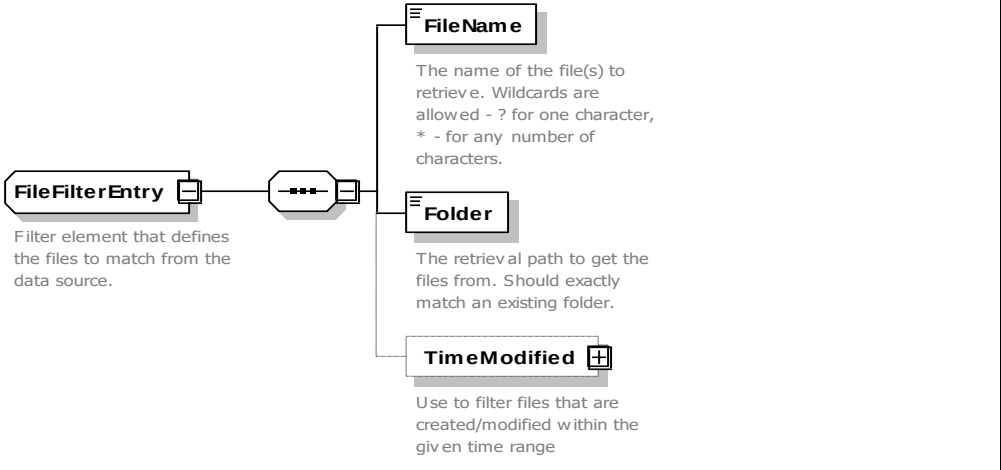
namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	
	pattern	[^\n\r\t`?\\<> :]*	

E.8 *simpleType TimeFormat*

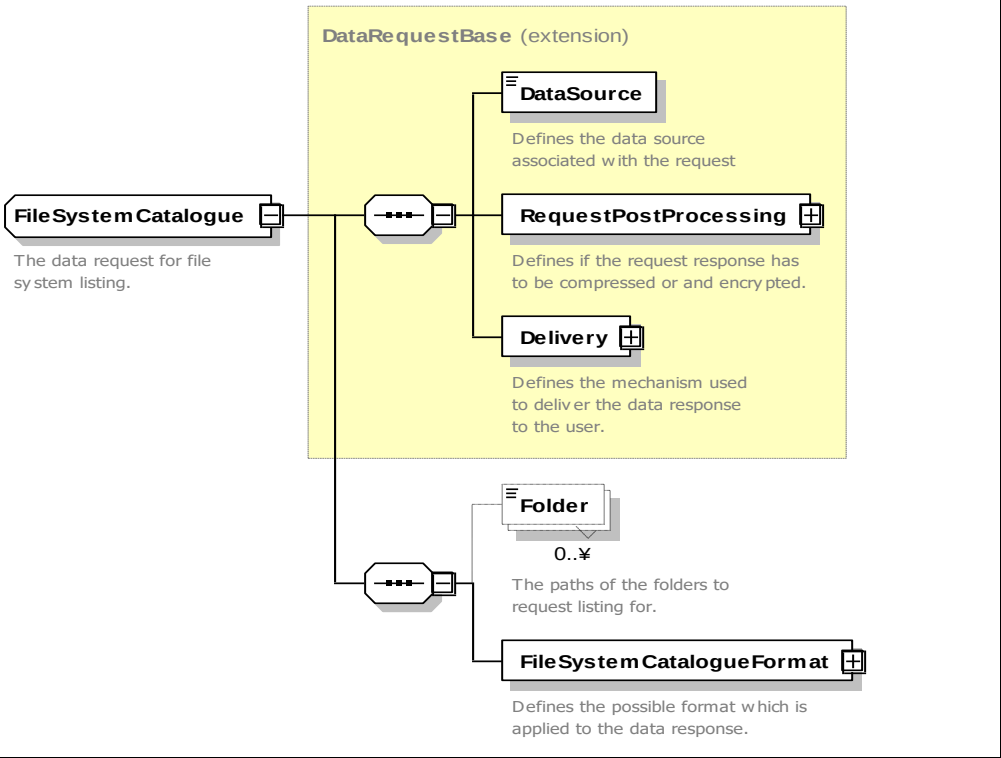
namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	
	enumeration	CCSDS	
	enumeration	DOY	
	enumeration	CUSTOM	

Appendix FFile System Data Types

F.1 complexType FileFilterEntry

diagram	 The diagram shows the structure of the FileFilterEntry complex type. It consists of a FileFilterEntry element (represented by a rounded rectangle with a small square on the right) connected to a container element (represented by a hexagon with four dots). This container element is connected to three child elements: FileName (a rectangle with a list icon), Folder (a rectangle with a list icon), and TimeModified (a rectangle with a plus icon). Each child element has a descriptive text block below it. FileFilterEntry Filter element that defines the files to match from the data source. FileName The name of the file(s) to retrieve. Wildcards are allowed - ? for one character, * - for any number of characters. Folder The retrieval path to get the files from. Should exactly match an existing folder. TimeModified Use to filter files that are created/modified within the given time range
namespace	http://edds.egos.esa/model
children	FileName Folder TimeModified
used by	element FileSystemFile/FileFilterEntry
annotation	documentation Filter element that defines the files to match from the data source.

F.2 complexType FileSystemCatalogue

diagram	 The diagram shows the structure of the FileSystemCatalogue complex type. It consists of a FileSystemCatalogue element (represented by a rounded rectangle with a small square on the right) connected to two container elements (represented by hexagons with four dots). The first container element is connected to three child elements: DataSource (a rectangle with a list icon), RequestPostProcessing (a rectangle with a plus icon), and Delivery (a rectangle with a plus icon). These three child elements are grouped within a yellow box labeled 'DataRequestBase (extension)'. The second container element is connected to two child elements: Folder (a rectangle with a list icon) and FileSystemCatalogueFormat (a rectangle with a plus icon). Each child element has a descriptive text block below it. FileSystemCatalogue The data request for file system listing. DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. Folder 0..4 The paths of the folders to request listing for. FileSystemCatalogueFormat Defines the possible format which is applied to the data response.
---------	--

namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase abstract true
children	DataSource RequestPostProcessing Delivery Folder FileSystemCatalogueFormat
used by	complexType FileSystemFileCatalogue FileSystemFolderCatalogue
annotation	documentation The data request for file system listing.

F.3 complexType *FileSystemCatalogueFormat*

diagram	The possible formatting options for file system catalogue response
namespace	http://edds.egos.esa/model
children	XML ASCII
used by	element FileSystemCatalogue/FileSystemCatalogueFormat
annotation	documentation The possible formatting options for file system catalogue response

F.4 complexType *FileSystemFile*

diagram	
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase

properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery FileFilterEntry FileSystemDelivery
annotation	documentation The data request of the files from the file system data source.

F.5 *complexType* **FileSystemFileCatalogue**

diagram	The diagram illustrates the structure of the FileSystemFileCatalogue complex type. It is an extension of FileSystemCatalogue. The main component is FileSystemFileCatalogue, which contains a list of FileSystemCatalogueFormat elements. This list is further divided into two groups: one for DataSource and RequestPostProcessing, and another for Delivery and Folder. The DataSource and RequestPostProcessing elements are associated with the Folder element. The Delivery and Folder elements are associated with the FileSystemCatalogueFormat element. The Folder element is also associated with the FileSystemCatalogueFormat element. The Folder element is also associated with the FileSystemCatalogueFormat element.
namespace	http://edds.egos.esa/model
type	extension of FileSystemCatalogue
properties	base FileSystemCatalogue
children	DataSource RequestPostProcessing Delivery Folder FileSystemCatalogueFormat
annotation	documentation The data request for file system listing of files in requested catalogues.

F.6 complexType FileSystemFileCatalogueElement

diagram	
namespace	http://edds.egos.esa/model
children	Folder FileName Modified Size
used by	elements FileSystemFileCatalogueElement FileSystemFileCatalogueResponse/FileSystemFileCatalogueElement
annotation	documentation Describes the single catalogue entry

F.7 complexType FileSystemFileCatalogueResponse

diagram	
namespac e	http://edds.egos.esa/model
children	FileSystemFileCatalogueElement
annotation	documentation Defines the response list of catalogue entries

F.8 complexType FileSystemFolderCatalogue

diagram	The diagram shows the structure of the FileSystemFolderCatalogue complex type. It is an extension of FileSystemCatalogue. The structure is as follows: FileSystemFolderCatalogue (extension) DataSource: Defines the data source associated with the request. RequestPostProcessing: Defines if the request response has to be compressed or and encrypted. Delivery: Defines the mechanism used to deliver the data response to the user. Folder (0..N): The paths of the folders to request listing for. FileSystemCatalogueFormat: Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of FileSystemCatalogue
properties	base FileSystemCatalogue
children	DataSource RequestPostProcessing Delivery Folder FileSystemCatalogueFormat
annotation	documentation The data request for file system structure recursively listing all subfolders

F.9 complexType FileSystemFolderCatalogueElement

diagram	The diagram shows the structure of the FileSystemFolderCatalogueElement complex type. It contains a single child element: Folder: Defines the name of the catalogue.
namespace	http://edds.egos.esa/model
children	Folder
used by	elements FileSystemFolderCatalogueElement FileSystemFolderCatalogueResponse/FileSystemFolderCatalogueElement
annotation	documentation Describes the single structure response entry

F.10 complexType FileSystemFolderCatalogueResponse

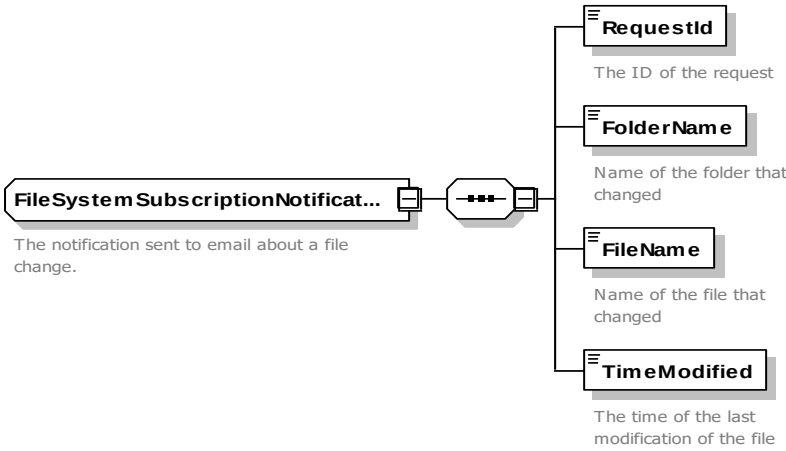
diagram	The diagram shows the structure of the FileSystemFolderCatalogueResponse complex type. It contains a single child element: FileSystemFolderCatalogueElement (0..N): Defines the response list of structure entries.
---------	--

namespace	http://edds.egos.esa/model
children	FileSystemFolderCatalogueElement
annotation	documentation Defines the response list of structure entries

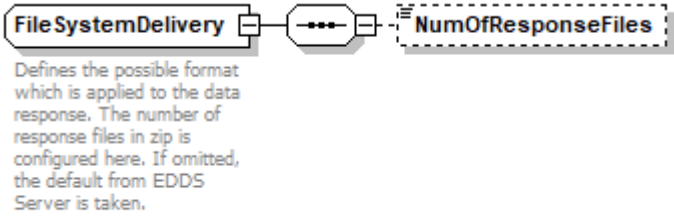
F.11 complexType *FileSystemSubscription*

diagram	File System Subscription Filter element that defines the files to match from the data source. DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. File Name The name of the file(s) to retrieve. Wildcards are allowed - ? for one character, * - for any number of characters. Folder The retrieval path to monitor the files from. Should exactly match an existing folder. Retrieve File If true, the file(s) will be automatically retrieved when the notification is received Send Email If true, an e-mail will be sent when the notification is received
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery File Name Folder RetrieveFile SendEmail
annotation	documentation Filter element that defines the files to match from the data source.

F.12 complexType FileSystemSubscriptionNotification

diagram	
namespace	http://edds.egos.esa/model
children	RequestId FolderName FileName TimeModified
used by	element FileSystemSubscriptionNotification
annotation	documentation The notification sent to email about a file change.

F.13 complexType FileSystemDelivery

diagram	
namespace	http://edds.egos.esa/model
children	NumOfResponseFiles
used by	element FileSystemFile/FileSystemDelivery
annotation	documentation Defines the possible format which is applied to the data response. The number of response files in zip is configured here. If omitted, the default from EDDS Server is taken.

Appendix G Filter Data Type

This section of the document describes the filter complex types as defined in the filter.xsd file.

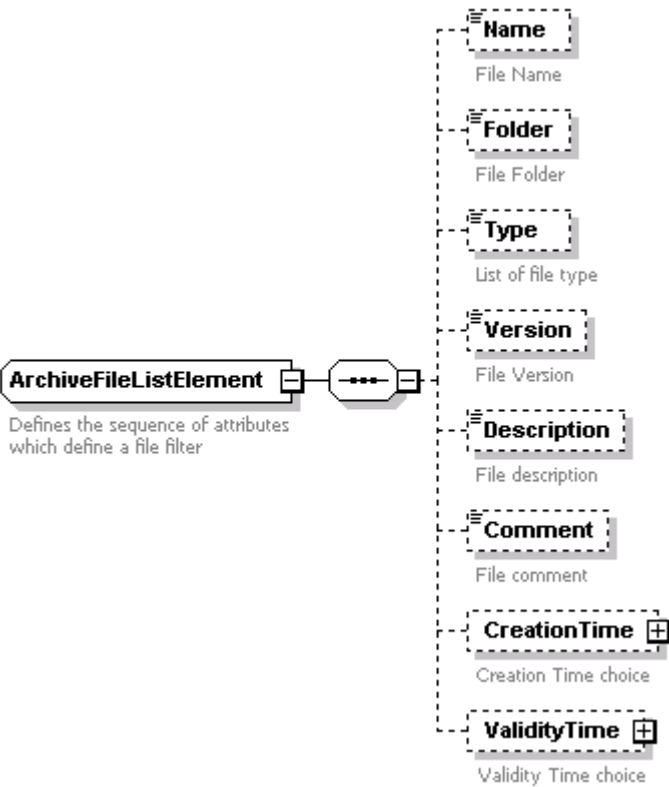
G.1 *complexType ArchiveFileFilter*

diagram	Defines the File Archive filter Defines the list of attributes to be used for the data filter
namespace	http://edds.egos.esa/model
children	ArchiveFileList
annotation	documentation Defines the File Archive filter

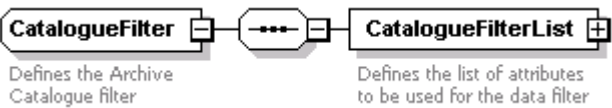
G.2 *complexType ArchiveFileList*

diagram	A list of archive files 0..∞ Defines the file filter element
namespace	http://edds.egos.esa/model
children	ArchiveFileListElement
used by	element ArchiveFileFilter/ArchiveFileList
annotation	documentation A list of archive files

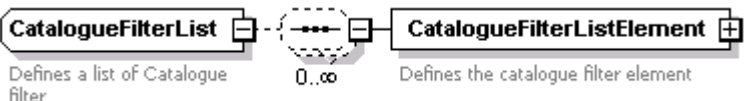
G.3 complexType ArchiveFileListElement

diagram	 ArchiveFileListElement Defines the sequence of attributes which define a file filter Name File Name Folder File Folder Type List of file type Version File Version Description File description Comment File comment CreationTime + Creation Time choice ValidityTime + Validity Time choice
namespace	http://edds.egos.esa/model
children	Name Folder Type Version Description Comment CreationTime ValidityTime
used by	element ArchiveFileList/ArchiveFileListElement
annotation	documentation Defines the sequence of attributes which define a file filter

G.4 complexType CatalogueFilter

diagram	 CatalogueFilter Defines the Archive Catalogue filter CatalogueFilterList + Defines the list of attributes to be used for the data filter
namespace	http://edds.egos.esa/model
children	CatalogueFilterList
annotation	documentation Defines the Archive Catalogue filter

G.5 complexType CatalogueFilterList

diagram	 CatalogueFilterList Defines a list of Catalogue filter CatalogueFilterListElement + Defines the catalogue filter element
namespace	http://edds.egos.esa/model
children	CatalogueFilterListElement
used by	element CatalogueFilter/CatalogueFilterList
annotation	documentation Defines a list of Catalogue filter

G.6 complexType CatalogueFilterListElement

diagram	
namespace	http://edds.egos.esa/model
children	Name Folder Type Version Description Comment CreationTime ValidityTime
used by	element CatalogueFilterList/CatalogueFilterListElement
annotation	documentation Defines the sequence of attributes which define a catalogue filter filter

G.7 complexType EddsUsageReportFilter

diagram	
namespace	http://edds.egos.esa/model
children	TimeRange EddsUsageReportFilterElement State AllUsersData OrderBy
used by	element EddsUsageReport/EddsUsageReportFilter
annotation	documentation Defines the sequence of attributes which define an Edds usage Report filter

namespace	http://edds.egos.esa/model
children	TimeRange EddsUsageReportFilterElement State AllUsersData OrderBy
annotation	documentation Defines the sequence of attributes which define an Edds usage Report filter

G.8 complexType EddsUsageReportFilterElement

diagram	
namespace	http://edds.egos.esa/model
children	EddsUsageReportFilterKeyword Operation
used by	element EddsUsageReportFilter/EddsUsageReportFilterElement

G.9 complexType EddsUsageReportFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	Role SubmissionTime SubmissionDOYTime ExecutionTime ExecutionDOYTime CompletionTime CompletionDOYTime ResponseDataSize ReasonOfFailure
used by	element EddsUsageReportFilterElement/EddsUsageReportFilterKeyword

G.10 complexType EgscActivityFilter

diagram	
namespace	http://edds.egos.esa/model
children	EgscActivityFilter/ActivityTimeFiltering

G.11 complexType EgsccParamDefinitionFilter

diagram	
namespace	http://edds.egos.esa/model
children	DataSpace
used by	EgsccParamDefinition/EgsccParamDefinitionFilter

G.12 complexType EgsccParamTmFilter

diagram	
namespace	http://edds.egos.esa/model
children	DataSpace ParamNameList TimeRange TimeFiltering DeliveryRange OnChange ParamFilterList SuperCommutationFlag MaxDecimalPlaces TdrsUseDOY TdrsStatsOnly TdrsDataOnly RepresentationSelection
used by	EgsccParam/EgsccParamTmFilter

G.13 complexType EgsccPktTcFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	MaximumDataVolume
used by	EgsccTcFilterElement/EgsccPktTcFilterKeyword

G.14 complexType EgsccPktTmFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	MaximumDataVolume
used by	EgsccTmFilterElement/EgsccPktTmFilterKeyword

G.15 complexType EgsccTcFilterElement

diagram	
namespace	http://edds.egos.esa/model
children	EgsccPktTcFilterKeyword Operation
used by	EgsccTcFilterList/EgsccTcFilterListElement

G.16 complexType EgsccTcFilterList

diagram	
namespace	http://edds.egos.esa/model
children	EgsccTcFilterListElement
used by	EgsccTcPacketFilter/EgsccTcFilterList

G.17 complexType EgsccTcPacketFilter

diagram	
namespace	http://edds.egos.esa/model

children	PacketNameList TimeRange TimeFiltering EgscdTmFilterList Dataspace BriefReportFlag IncludeRawData EgscdTcFilterList
used by	EgscnPktTcReport/EgscdTcPacketFilter

G.18 complexType *EgscdTmFilterElement*

diagram	
namespace	http://edds.egos.esa/model
children	EgscnPktTmFilterKeyword Operation
used by	EgscdTmFilterList/EgscdTmFilterListElement

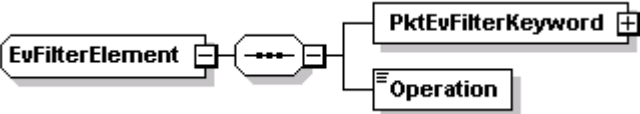
G.19 complexType *EgscdTmFilterList*

diagram	
namespace	http://edds.egos.esa/model
children	EgscdTmFilterListElement
used by	EgscdTmPacketFilter/EgscdTmFilterList

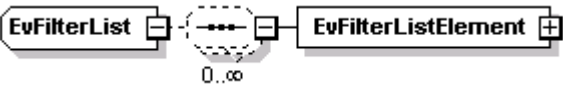
G.20 complexType *EgscdTmPacketFilter*

diagram	
namespace	http://edds.egos.esa/model
children	PacketNameList TimeRange TimeFiltering EgscdTmFilterList Dataspace BriefReportFlag IncludeRawData
used by	EgscnPktTmReport/EgscdTmPacketFilter EgscnPktTm/EgscdTmPacketFilter EgscnPktTmRaw/EgscdTmPacketFilter EgscnPktTmStatistics/EgscdTmPacketFilter EgscdTcPacketFilter

G.21 complexType *EvFilterElement*

diagram	
namespace	http://edds.egos.esa/model
children	PktEvFilterKeyword Operation
used by	element EvFilterList/EvFilterListElement

G.22 complexType *EvFilterList*

diagram	
namespace	http://edds.egos.esa/model
children	EvFilterListElement

used by	element EvPacketFilter/EvFilterList
---------	--

G.23 complexType EvPacketFilter

diagram	The diagram shows the structure of the EvPacketFilter complex type. It is represented by a rounded rectangle with a small square on the left and a dashed line with three dots in the middle. To its right is a vertical stack of four elements: TimeRange (with a plus sign in a square), TimeFiltering (with a list icon), EvFilterList (dashed box with a plus sign and $0..∞$ below it), and Dataspace (dashed box with a list icon and a description: "The dataspace to direct the request to. Current will be used if not included.").
namespace	http://edds.egos.esa/model
children	TimeRange TimeFiltering EvFilterList Dataspace

G.24 complexType OolDataReportFilter

diagram	The diagram shows the structure of the OolDataReportFilter complex type. It is represented by a rounded rectangle with a small square on the left and a dashed line with three dots in the middle. To its right is a vertical stack of four elements: TimeRange (with a plus sign in a square), TimeFiltering (with a list icon and a description: "Defines the type of time retrieval"), OolFilterList (dashed box with a plus sign and $0..∞$ below it), and Dataspace (dashed box with a list icon and a description: "The dataspace to direct the request to. Current will be used if not included.").
namespace	http://edds.egos.esa/model
children	TimeRange TimeFiltering OolFilterList Dataspace
annotation	documentation Defines the OOL data report filter

G.25 complexType OolFilterElement

diagram	The diagram shows the structure of the OolFilterElement complex type. It is represented by a rounded rectangle with a small square on the left and a dashed line with three dots in the middle. To its right is a vertical stack of two elements: OolFilterKeyword (with a plus sign in a square) and Operation (with a list icon).
namespace	http://edds.egos.esa/model

children	OolFilterKeyword Operation	
used by	element	OolFilterList/OolFilterListElement

G.26 complexType OolFilterKeyword

diagram	RecordType One of LIMIT, STATE, STATUS_CONSISTENCY, SOFT_HARD_LIMIT, LIMIT_MO, DELTA ParameterName ParameterState One of NOMINAL, WARNING, OUT_OF_LIMITS, SCC, SCC_DISABLE, SCC_OFF, SCC_UNINIT, VIOLATION OolState Value LowerLimit Only for LIMIT, SOFT_HARD_LIMIT, LIMIT_MO types UpperLimit Only for LIMIT, SOFT_HARD_LIMIT, LIMIT_MO types AllowedValues Each value should be comma separated SccLimit Only for STATUS_CONSISTENCY types ReferenceValue Only for DELTA types DeltaValue Only for DELTA types MinValue Only for DELTA types LastValue Only for DELTA types ViolationNumber Not for DELTA types ViolationNumberHard Only for SOFT_HARD_LIMIT types ViolationNumberSoft Only for SOFT_HARD_LIMIT types Description Unit MaximumDataVolume
namespace	http://edds.egos.esa/model
children	RecordType ParameterName ParameterState OolState Value LowerLimit UpperLimit AllowedValues SccLimit ReferenceValue DeltaValue MinValue LastValue ViolationNumber ViolationNumberHard ViolationNumberSoft Description Unit MaximumDataVolume

	ViolationNumberSoft Description Unit MaximumDataVolume
used by	element OolFilterElement/OolFilterKeyword

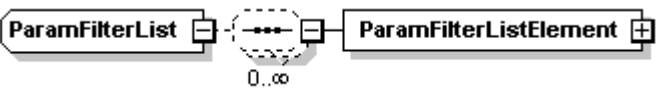
G.27 complexType OolFilterList

diagram	
namespace	http://edds.egos.esa/model
children	OolFilterListElement
used by	element OolDataReportFilter/OolFilterList

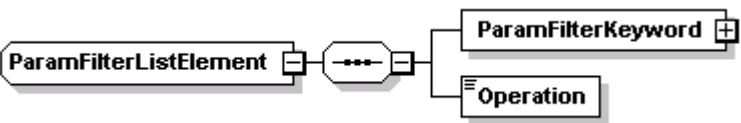
G.28 complexType ParamFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	ParamValidityStatus OnBoardTime OnBoardDOYTime EngineeringValue EngineeringValueString MaximumDataVolume RawValue RawValueString ParentId ParentGenTime ParentGenDOYTime
used by	element ParamFilterListElement/ParamFilterKeyword

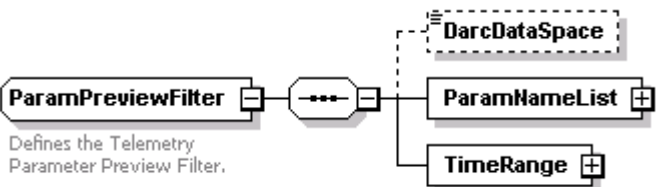
G.29 complexType ParamFilterList

diagram	
namespace	http://edds.egos.esa/model
children	ParamFilterListElement
used by	element ParamTmFilter/ParamFilterList

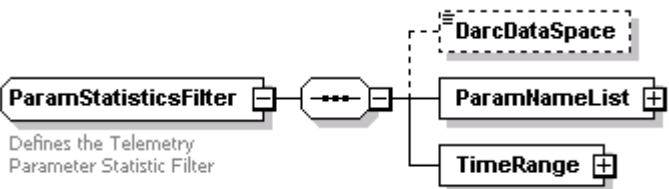
G.30 complexType ParamFilterListElement

diagram	
namespace	http://edds.egos.esa/model
children	ParamFilterKeyword Operation
used by	element ParamFilterList/ParamFilterListElement

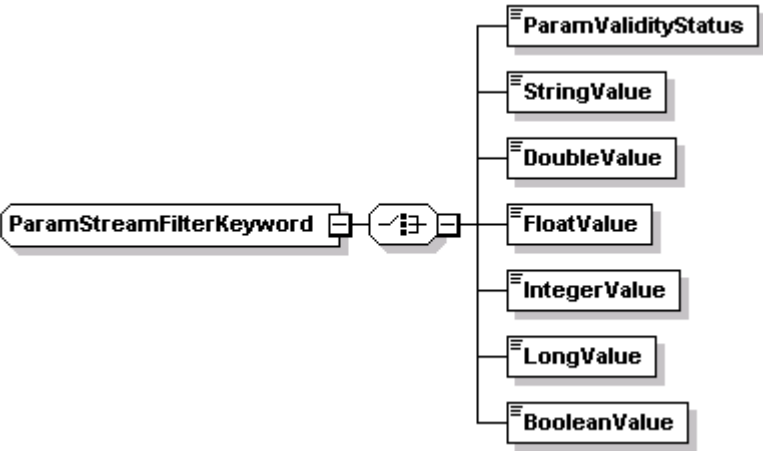
G.31 complexType ParamPreviewFilter

diagram	
namespace	http://edds.egos.esa/model
children	DarcDataSpace ParamNameList TimeRange
annotation	documentation Defines the Telemetry Parameter Preview Filter.

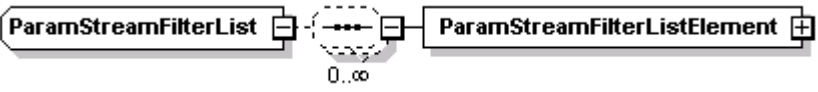
G.32 complexType ParamStatisticsFilter

diagram	
namespace	http://edds.egos.esa/model
children	DarcDataSpace ParamNameList TimeRange
annotation	documentation Defines the Telemetry Parameter Statistic Filter

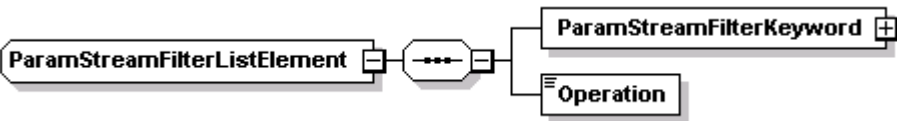
G.33 complexType ParamStreamFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	ParamValidityStatus StringValue DoubleValue FloatValue IntegerValue LongValue BooleanValue
used by	element ParamStreamFilterListElement/ParamStreamFilterKeyword

G.34 complexType ParamStreamFilterList

diagram	
namespace	http://edds.egos.esa/model
children	ParamStreamFilterListElement

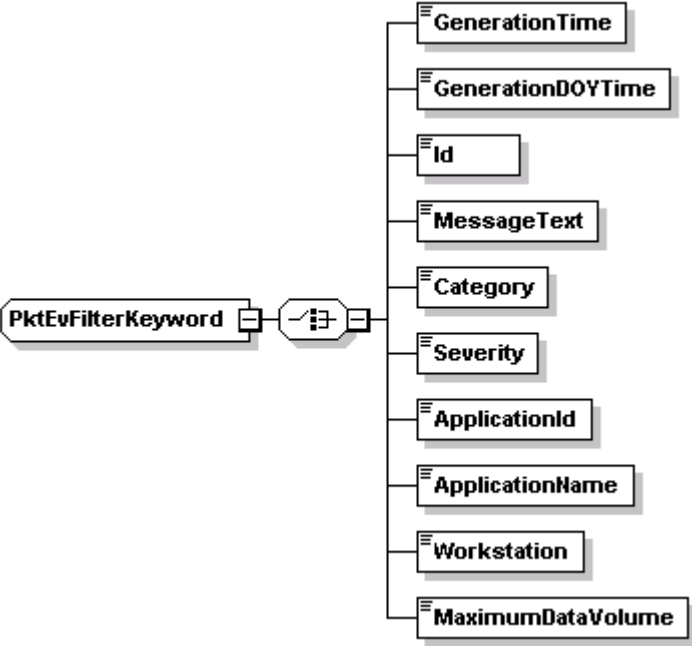
G.35 complexType ParamStreamFilterListElement

diagram	
namespace	http://edds.egos.esa/model
children	ParamStreamFilterKeyword Operation
used by	element ParamStreamFilterList/ParamStreamFilterListElement

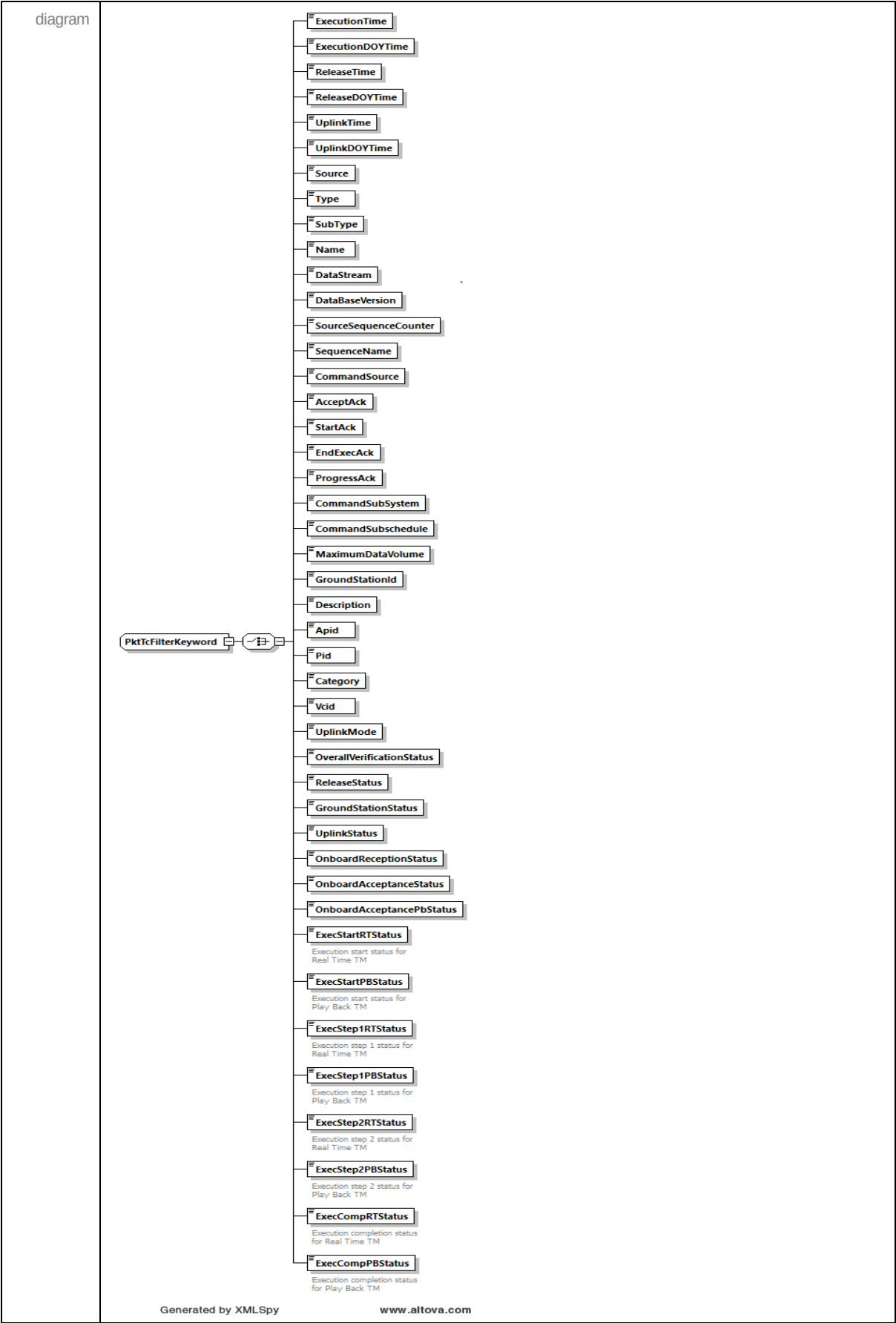
G.36 complexType ParamTmFilter

diagram	ParamTmFilter Defines the Telemetry Parameter Filter DarcDataSpace ParamNameList + TimeRange + TimeFiltering Defines the type of time retrieval. If omitted, the default of GENERATION_TIME is used. Note that if STORAGE_TIME is specified, this is the storage time of the parameter sample in the DARC. DeliveryRange + OnChange ParamFilterList + 0..∞ SuperCommutationFlag Not supported by DARC MaxDecimalPlaces The maximum number of decimal places for float-type data for the TDRS format. Default is 3. TdrsUseDOY If true, the TDRS format will use the DOY format for dates, otherwise it will use DMY. TdrsStatsOnly If true, only the statistics will be included the TDRS report output TdrsDataOnly If true, only the data will be included the TDRS report output RepresentationSelection
namespace	http://edds.egos.esa/model
children	DarcDataSpace ParamNameList TimeRange TimeFiltering DeliveryRange OnChange ParamFilterList SuperCommutationFlag MaxDecimalPlaces TdrsUseDOY TdrsStatsOnly TdrsDataOnly RepresentationSelection
annotation	documentation Defines the Telemetry Parameter Filter

G.37 complexType PktEvFilterKeyword

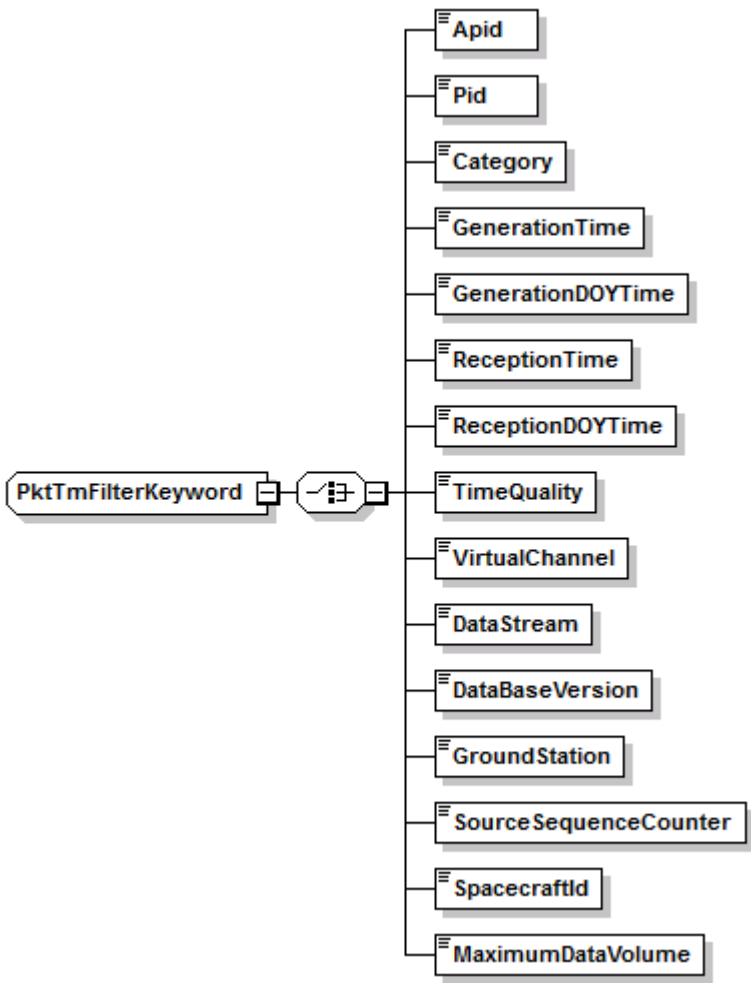
diagram	 The diagram shows a complex type PktEvFilterKeyword connected to a container element (a rectangle with a dashed border and a small icon). This container is connected to nine child elements, each in a box with a small icon: GenerationTime, GenerationDOYTime, Id, MessageText, Category, Severity, ApplicationId, ApplicationName, Workstation, and MaximumDataVolume.
namespace	http://edds.egos.esa/model
children	GenerationTime GenerationDOYTime Id MessageText Category Severity ApplicationId ApplicationName Workstation MaximumDataVolume
used by	element EvFilterElement/PktEvFilterKeyword

G.38 complexType PktTcFilterKeyword



namespace	http://edds.egos.esa/model
children	ExecutionTime ExecutionDOYTime ReleaseTime ReleaseDOYTime Type SubType Name DataStream DataBaseVersion SourceSequenceCounter SequenceName CommandSource AcceptAck StartAck EndExecAck ProgressAck CommandSubSystem CommandSubschedule MaximumDataVolume GroundStationId Description Apid UplinkMode OverallVerificationStatus ReleaseStatus GroundStationStatus UplinkStatus OnboardReceptionStatus OnboardAcceptanceStatus OnboardAcceptancePbStatus ExecStartRTStatus ExecStartPBStatus ExecStep1RTStatus ExecStep1PBStatus ExecStep2RTStatus ExecStep2PBStatus ExecCompRTStatus ExecCompPBStatus
used by	element TcFilterElement/PktTcFilterKeyword

G.39 complexType PktTmFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	Apid Pid Category GenerationTime GenerationDOYTime ReceptionTime ReceptionDOYTime TimeQuality VirtualChannel DataStream DataBaseVersion GroundStation SourceSequenceCounter SpacecraftId MaximumDataVolume
used by	element TmFilterElement/PktTmFilterKeyword

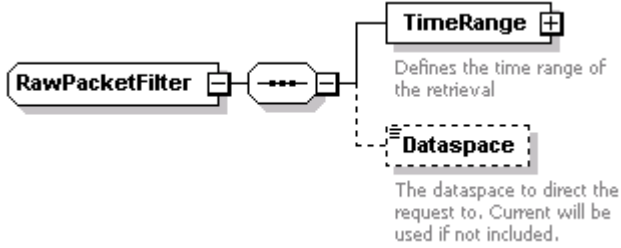
G.40 complexType PktTmGapReportFilter

diagram	The diagram shows the structure of the PktTmGapReportFilter complexType. It consists of a root element PktTmGapReportFilter which contains three child elements: PacketNameList, TimeRange, and DataPartition. The PacketNameList and TimeRange elements are shown as solid boxes, while DataPartition is shown as a dashed box. The DataPartition element has a value of 0..∞. The DataPartition element is also shown as a dashed box. The DataPartition element is also shown as a dashed box. PacketNameList + Defines the Packets which have to be retrieved TimeRange + Defines the time range of the retrieval DataPartition 0..∞ The data partitions to fetch the report for. Default specified in EDDS properties will be used if not included. Dataspace The dataspace to direct the request to. Current will be used if not included.
namespace	http://edds.egos.esa/model
children	PacketNameList TimeRange DataPartition Dataspace

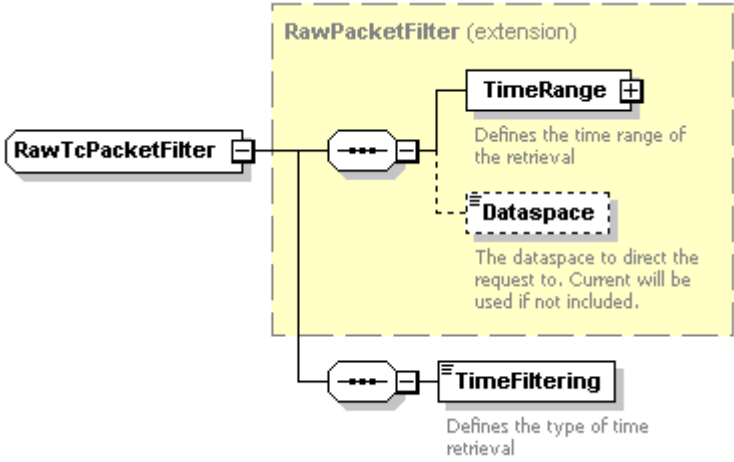
G.41 complexType RawEvPacketFilter

diagram	The diagram shows the structure of the RawEvPacketFilter complexType. It consists of a root element RawEvPacketFilter which contains three child elements: TimeRange, Dataspace, and TimeFiltering. The TimeRange and Dataspace elements are shown as solid boxes, while TimeFiltering is shown as a dashed box. The TimeRange element has a value of 0..∞. The Dataspace element is shown as a dashed box. The TimeFiltering element is shown as a dashed box. RawEvPacketFilter (extension) TimeRange + Defines the time range of the retrieval Dataspace The dataspace to direct the request to. Current will be used if not included. TimeFiltering Defines the type of time retrieval
namespace	http://edds.egos.esa/model
type	extension of RawPacketFilter
properties	base RawPacketFilter
children	TimeRange Dataspace TimeFiltering
used by	complexType RawTmPacketFilter

G.42 complexType RawPacketFilter

diagram	 The diagram shows the RawPacketFilter complexType structure. It consists of a TimeRange element and a Dataspace element. The TimeRange element is defined as "Defines the time range of the retrieval". The Dataspace element is defined as "The dataspace to direct the request to. Current will be used if not included.".
namespace	http://edds.egos.esa/model
properties	abstract true
children	TimeRange Dataspace
used by	complexType RawEvPacketFilter RawTcPacketFilter

G.43 complexType RawTcPacketFilter

diagram	 The diagram shows the RawTcPacketFilter complexType structure. It is an extension of RawPacketFilter. The RawPacketFilter (extension) part contains a TimeRange element and a Dataspace element. The TimeRange element is defined as "Defines the time range of the retrieval". The Dataspace element is defined as "The dataspace to direct the request to. Current will be used if not included.". The RawTcPacketFilter element also includes a TimeFiltering element, defined as "Defines the type of time retrieval".
namespace	http://edds.egos.esa/model
type	extension of RawPacketFilter
properties	base RawPacketFilter
children	TimeRange Dataspace TimeFiltering

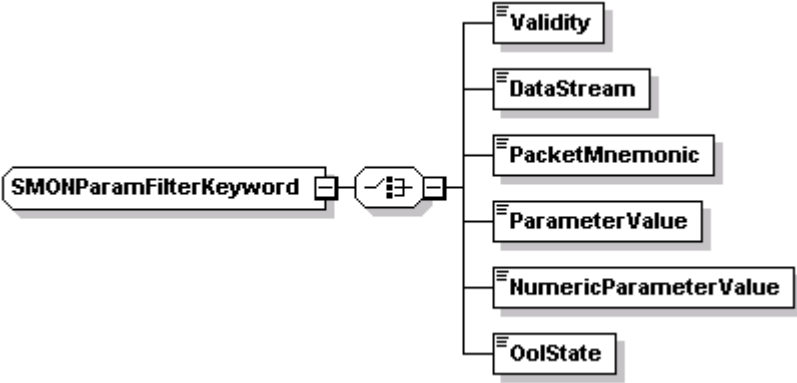
G.44 complexType RawTmPacketFilter

diagram	
namespace	http://edds.egos.esa/model
type	extension of RawEvPacketFilter
properties	base RawEvPacketFilter
children	TimeRange Dataspace TimeFiltering PacketNameList
annotation	documentation Defines the extra filter elements needed for TM packets

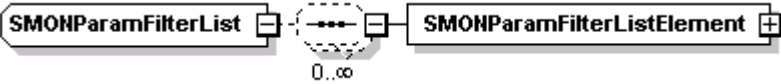
G.45 complexType SMONParamFilter

diagram	
namespace	http://edds.egos.esa/model
children	ParamNameList TimeRange SMONParamFilterList
used by	element SMONParam/SMONParamFilter
annotation	documentation Defines the SMON Parameter Filter

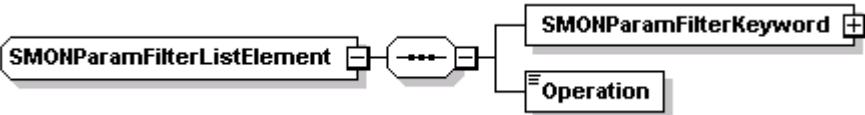
G.46 complexType SMONParamFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	Validity DataStream PacketMnemonic ParameterValue NumericParameterValue OolState
used by	element SMONParamFilterListElement/SMONParamFilterKeyword

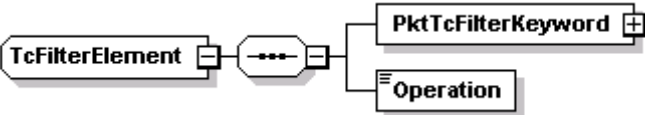
G.47 complexType SMONParamFilterList

diagram	
namespace	http://edds.egos.esa/model
children	SMONParamFilterListElement
used by	element SMONParamFilter/SMONParamFilterList

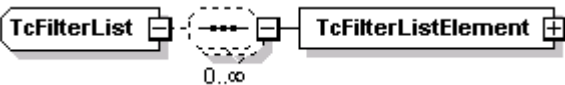
G.48 complexType SMONParamFilterListElement

diagram	
namespace	http://edds.egos.esa/model
children	SMONParamFilterKeyword Operation
used by	element SMONParamFilterList/SMONParamFilterListElement

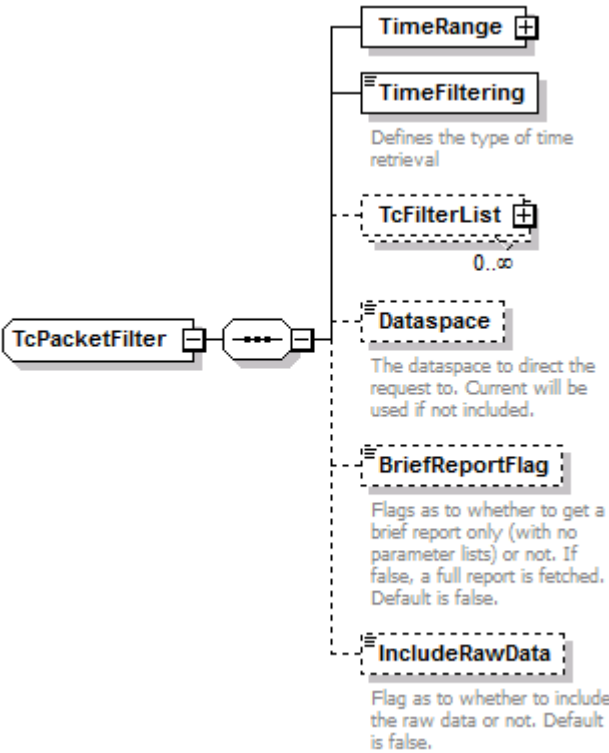
G.49 complexType TcFilterElement

diagram	
namespace	http://edds.egos.esa/model
children	PktTcFilterKeyword Operation
used by	element TcFilterList/TcFilterListElement

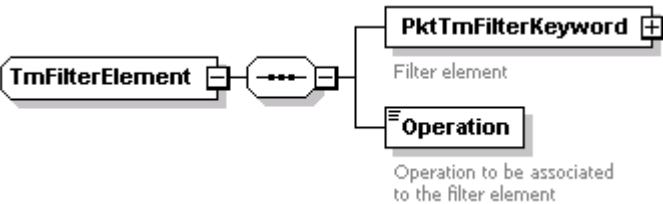
G.50 complexType TcFilterList

diagram	
namespace	http://edds.egos.esa/model
children	TcFilterListElement
used by	element TcPacketFilter/TcFilterList

G.51 complexType TcPacketFilter

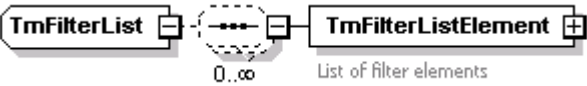
diagram	
namespace	http://edds.egos.esa/model
children	TimeRange TimeFiltering TcFilterList Dataspace BriefReportFlag IncludeRawData

G.52 complexType TmFilterElement

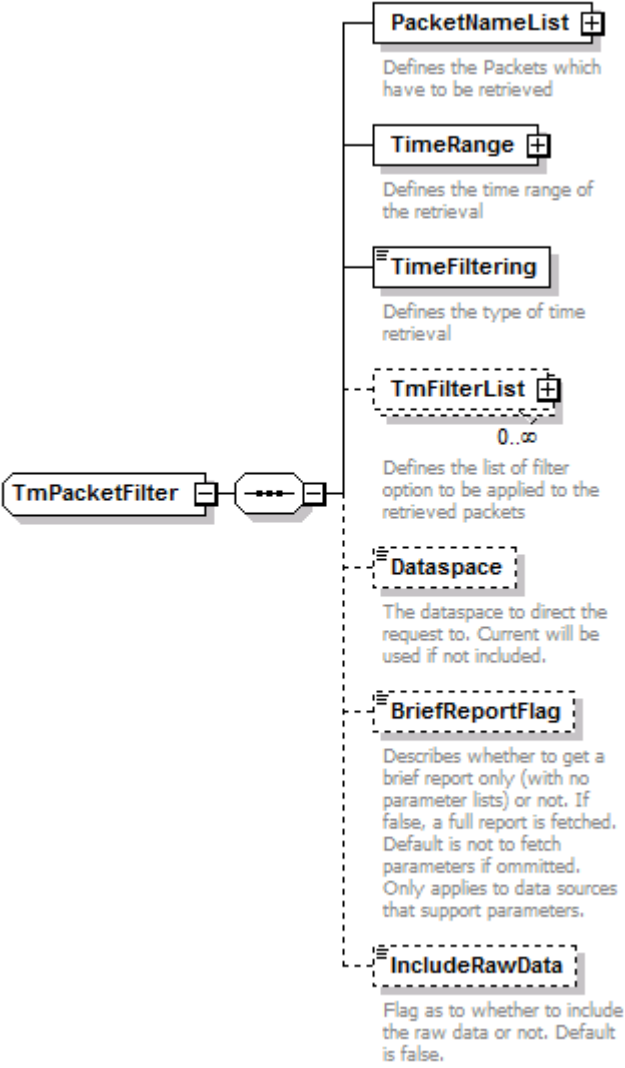
diagram	
---------	--

namespace	http://edds.egos.esa/model
children	PktTmFilterKeyword Operation
used by	element TmFilterList/TmFilterListElement

G.53 complexType TmFilterList

diagram	
namespace	http://edds.egos.esa/model
children	TmFilterListElement
used by	element TmPacketFilter/TmFilterList

G.54 complexType TmPacketFilter

diagram	
namespace	http://edds.egos.esa/model

children	PacketNameList TimeRange TimeFiltering TmFilterList Dataspace BriefReportFlag IncludeRawData
----------	---

G.55 simpleType CommandSource

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
used by	element	PktTcFilterKeyword/CommandSource
facets	Kind	Value
	enumeration	MANUAL_STACK
	enumeration	AUTO_STACK
	enumeration	EXT_SOURCE
	enumeration	TC_SPACON
	enumeration	OBQM_DISP
	enumeration	SMF_SOURCE
	enumeration	NAVPRS_STACK
	enumeration	MKMF_STACK
	enumeration	PKMF_STACK
	enumeration	SKMF_STACK

G.56 simpleType RepresentationSelection

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
used by	element	SMONParamFilter
facets	Kind	Value
	enumeration	ENG
	enumeration	RAW
	enumeration	ALL

G.57 simpleType DarcRepresentationSelection

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
used by	element	ParamTmFilter
facets	Kind	Value
	enumeration	ENG
	enumeration	RAW

G.58 simpleType StageStatus

namespace	http://edds.egos.esa/model	
type	restriction of xs:string	

properties	base xs:string		
used by	elements	PktTcFilterKeyword/ExecCompPBStatus PktTcFilterKeyword/ExecCompRTStatus PktTcFilterKeyword/ExecStartPBStatus PktTcFilterKeyword/ExecStartRTStatus PktTcFilterKeyword/ExecStep1PBStatus PktTcFilterKeyword/ExecStep1RTStatus PktTcFilterKeyword/ExecStep2PBStatus PktTcFilterKeyword/ExecStep2RTStatus PktTcFilterKeyword/GroundStationStatus PktTcFilterKeyword/OnboardAcceptanceStatus PktTcFilterKeyword/OnboardAcceptancePbStatus PktTcFilterKeyword/OnboardReceptionStatus PktTcFilterKeyword/OverallVerificationStatus PktTcFilterKeyword/ReleaseStatus PktTcFilterKeyword/UplinkStatus	
facets	Kind	Value	annotation
	enumeration		documentation Not applicable
	enumeration	I	documentation I for idle
	enumeration	P	documentation P for pending
	enumeration	S	documentation S for success
	enumeration	F	documentation F for failed
	enumeration	U	documentation U for unverified
	enumeration	X	documentation X for unknown
	enumeration	T	documentation T for timeout
	enumeration	E	documentation E for superseded
	enumeration	N	documentation N for uncertain failed
	enumeration	V	documentation V for uncertain successful
	enumeration	C	documentation C for affected
	enumeration	D	documentation D for disabled
	enumeration	A	documentation A for assumed passed
	enumeration	SCC	documentation SCC for Status Consistency Check

G.59 simpleType TcTimeFiltering

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
used by	elements	TcPacketFilter/TimeFiltering RawTcPacketFilter/TimeFiltering	
facets	Kind	Value	annotation
	enumeration	RELEASE_TIME	documentation Command release time of the packet (Primary Key)
	enumeration	EXECUTION_TIME	documentation Predicted or actual execution time of the packet (Secondary Key)

G.60 simpleType ActivityTimeFiltering

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
used by	elements	EgscActivityFilter	

facets	Kind	Value	annotation
	enumeration	GENERATION_TIME	documentation The time when a given data unit is created by its source.
	enumeration	EXECUTION_TIME	documentation The time when a data unit is received by the EGS-CC system.

G.61 simpleType TimeFiltering

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	elements	EvPacketFilter/TimeFiltering OolDataReportFilter/TimeFiltering TmPacketFilter/TimeFiltering RawEvPacketFilter/TimeFiltering	
facets	Kind	Value	annotation
	enumeration	GENERATION_TIME	documentation On board Time of the packet (Primary Key)
	enumeration	STORAGE_TIME	documentation Reception time of the packet into the archive (Secondary Key)

G.62 simpleType UplinkMode

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	elements	PktTcFilterKeyword/UplinkMode	
facets	Kind	Value	annotation
	enumeration	A	documentation AD transmission mode
	enumeration	B	documentation BD transmission mode

G.63 simpleType Severity

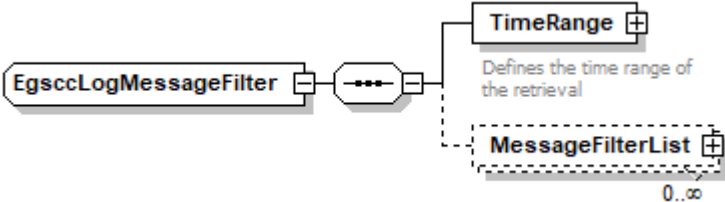
namespace	http://edds.egos.esa/model		
type	Severity		
type	restriction of xs:string		
properties	base	xs:string	
used by	elements	PktEvFilterKeyword/Severity	
facets	Kind	Value	annotation
	enumeration	Warning	
	enumeration	Error	
	enumeration	Information	
facets	enumeration	Fatal	

G.64 simpleType Category

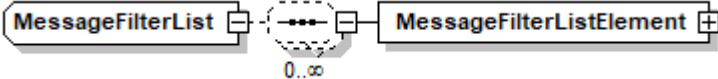
namespace	http://edds.egos.esa/model		
type	Category		
type	restriction of xs:string		
properties	base	xs:string	
used by	elements	PktEvFilterKeyword/Category	

facets	Kind	Value	annotation
	enumeration	Log	
	enumeration	Software	
	enumeration	System	
	enumeration	MIB	

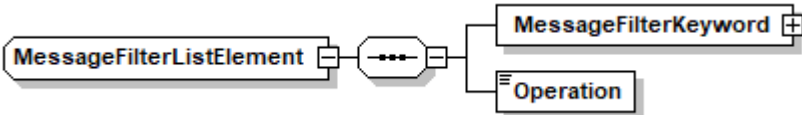
G.65 complexType EgscclLogMessageFilter

diagram	
namespace	http://edds.egos.esa/model
children	TimeRange MessageFilterList
used by	EgscclLogMessageReport/EgscclLogMessageFilter

G.66 complexType MessageFilterList

diagram	
namespace	http://edds.egos.esa/model
children	MessageFilterListElement
used by	EgscclLogMessageFilter/MessageFilterList

G.67 complexType MessageFilterListElement

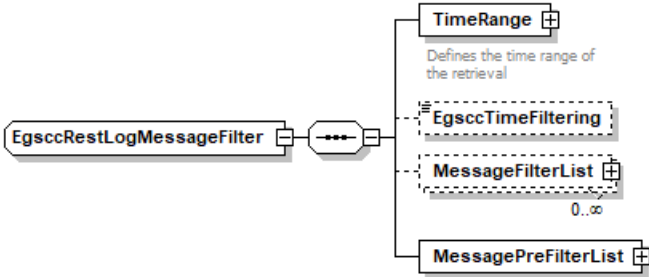
diagram	
namespace	http://edds.egos.esa/model
children	MessageFilterKeyword Operation
used by	MessageFilterList/ MessageFilterListElement

G.68 complexType MessageFilterKeyword

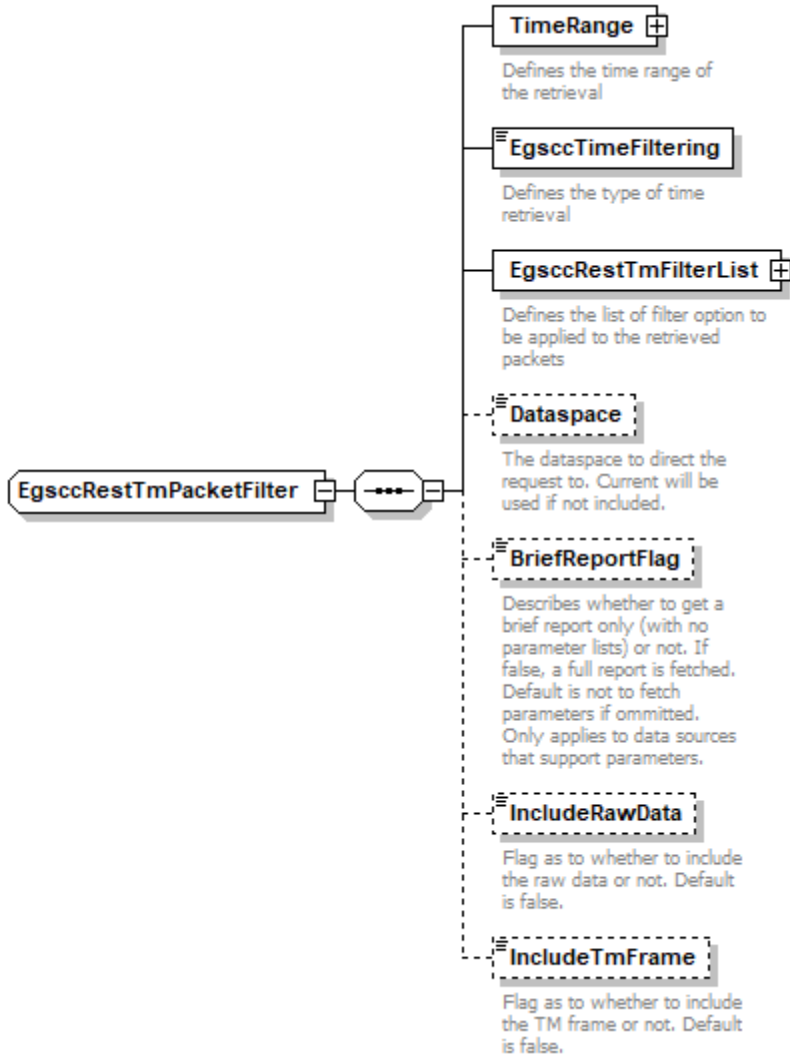
diagram	
namespace	http://edds.egos.esa/model
children	messageId logbookId nodeId generationTime egscReceptionTime storageTime categoryId applicationId severity systemId messageText messageType annotation contextId systemSessionId createdBy originName originAddress acknowledgementType acknowledgementTime requiresAcknowledgement acknowledgementUser traceScope traceLevel

used by	MessageFilterListElement/ MessageFilterKeyword
---------	--

G.69 complexType EgsccRestLogMessageFilter

diagram	
namespace	http://edds.egos.esa/model
children	TimeRange EgsccTimeFiltering MessageFilterList MessagePreFilterList
used by	element EgsccRestLogMessageReport/EgsccRestLogMessageFilter

G.70 complexType EgsccRestTmPacketFilter

diagram	
namespace	http://edds.egos.esa/model

children	TimeRange EgscTimeFiltering EgscRestTmFilterList Dataspace BriefReportFlag IncludeRawData IncludeTmFrame
used by	element EgscRestPktTmReport/EgscRestTmPacketFilter EgscRestPktTm/EgscRestTmPacketFilter

G.71 complexType EgscRestTcPacketFilter

diagram	The diagram shows the structure of the EgscRestTcPacketFilter complex type. It consists of a root element EgscRestTcPacketFilter (rectangle with a small square on the left) connected to a sequence container (octagon with three dots). This container is connected to five child elements: TimeRange (rectangle with a plus icon), EgscTimeFiltering (rectangle with a plus icon), EgscRestTcFilterList (rectangle with a plus icon), Dataspace (dashed rectangle), BriefReportFlag (dashed rectangle), and IncludeRawData (dashed rectangle). Each element has a descriptive text below it: TimeRange defines the time range of the retrieval; EgscTimeFiltering defines the type of time retrieval; EgscRestTcFilterList defines the list of filter options to be applied to the retrieved packets; Dataspace is the dataspace to direct the request to, with 'Current' used if not included; BriefReportFlag describes whether to get a brief report only (with no parameter lists) or not, with a default of 'not to fetch parameters if omitted' and applying only to data sources that support parameters; IncludeRawData is a flag as to whether to include the raw data or not, with a default of 'false'.
namespace	http://edds.egos.esa/model
children	TimeRange EgscTimeFiltering EgscRestTcFilterList Dataspace BriefReportFlag IncludeRawData
used by	element EgscRestPktTcReport/EgscRestTcPacketFilter

G.72 complexType EgscRestTcFilterList

diagram	The diagram shows the structure of the EgscRestTcFilterList complex type. It consists of a root element EgscRestTcFilterList (rectangle with a small square on the left) connected to a sequence container (octagon with three dots). This container is connected to one child element: EgscRestTcFilterListElement (rectangle with a plus icon). Below the container is the cardinality 0..∞ and the text List of filter elements.
namespace	http://edds.egos.esa/model
children	EgscRestTcFilterListElement
used by	element EgscRestTcPacketFilter/EgscRestTcFilterList

G.73 complexType EgscRestTcFilterElement

diagram	The diagram shows the structure of the EgscRestTcFilterElement complex type. It consists of a root element EgscRestTcFilterElement (rectangle with a small square on the left) connected to a sequence container (octagon with three dots). This container is connected to two child elements: EgscRestPktTcFilterKeyword (rectangle with a plus icon) and Operation (rectangle with a plus icon). Below the container is the text Filter element and Operation to be associated to the filter element.
---------	--

namespace	http://edds.egos.esa/model
children	EgscRestPktTcFilterKeyword Operation
used by	element EgscRestTcFilterList/EgscRestTcFilterListElement

G.74 complexType EgscRestPktTcFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	Apid Live PacketId PacketInstancelId PusSubType PusType QualityFlag SourceSequenceCounter SpacecraftId VirtualChannelId ActivityOccurrence ActivityRoute
used by	element EgscRestTcFilterElement/EgscRestPktTcFilterKeyword

G.75 complexType EgsccParamTmFilter

diagram	EgsccParamTmFilter Defines the Telemetry Parameter Filter for egsc DataSpace ParamNameList TimeRange TimeFiltering Defines the type of time retrieval. If omitted, the default of GENERATION_TIME is used. Note that if STORAGE_TIME is specified, this is the storage time of the parameter sample in the DARC. DeliveryRange OnChange ParamFilterList 0..∞ SuperCommutationFlag Not supported by DARC MaxDecimalPlaces The maximum number of decimal places for float-type data for the TDRS format. Default is 3. TdrsUseDOY If true, the TDRS format will use the DOY format for dates, otherwise it will use DMY. TdrsStatsOnly If true, only the statistics will be included the TDRS report output TdrsDataOnly If true, only the data will be included the TDRS report output RepresentationSelection
namespace	http://edds.egos.esa/model
children	DataSpace ParamNameList TimeRange TimeFiltering DeliveryRange OnChange ParamFilterList SuperCommutationFlag MaxDecimalPlaces TdrsUseDOY TdrsStatsOnly TdrsDataOnly RepresentationSelection
used by	elements EgsccRestParam/EgsccParamTmFilter EgsccParam/EgsccParamTmFilter

G.76 complexType EgsccRestActivityFilter

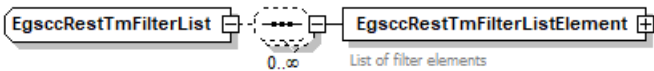
diagram	
namespace	http://edds.egos.esa/model
children	TimeRange Dataspace EgsccTimeFiltering EgsccRestActivityFilterList
used by	element EgsccRestActivityReport/EgsccRestActivityFilter

G.77 complexType EgsccRestParamTmFilter

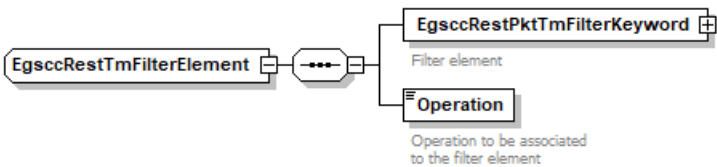
diagram	
---------	--

namespace	http://edds.egos.esa/model
children	DataSpace TimeRange EgscctimeFiltering EgsccrestParamFilterList DeliveryRange OnChange SuperCommutationFlag MaxDecimalPlaces TdrsUseDOY TdrsStatsOnly TdrsDataOnly RepresentationSelection
used by	element EgsccrestParam/EgsccrestParamTmFilter

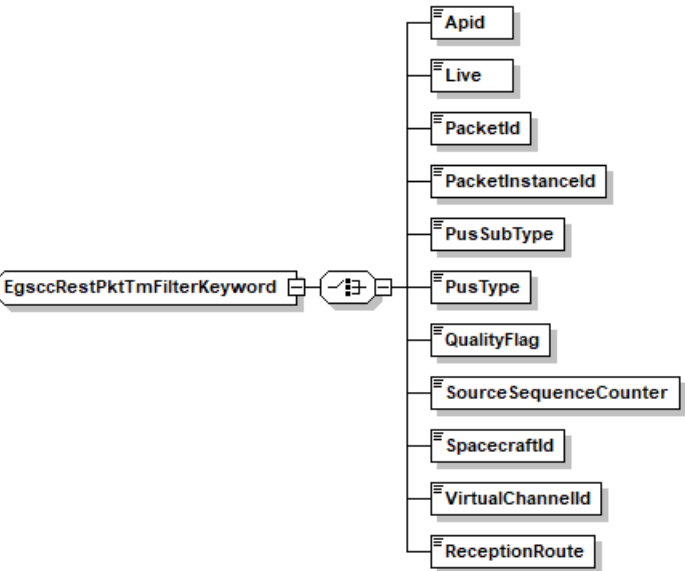
G.78 complexType EgsccrestTmFilterList

diagram	
namespace	http://edds.egos.esa/model
children	EgsccrestTmFilterListElement
used by	element EgsccrestTmPacketFilter/EgsccrestTmFilterList

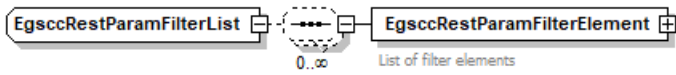
G.79 complexType EgsccrestTmFilterElement

diagram	
namespace	http://edds.egos.esa/model
children	EgsccrestPktTmFilterKeyword Operation
used by	element EgsccrestTmFilterList/EgsccrestTmFilterListElement

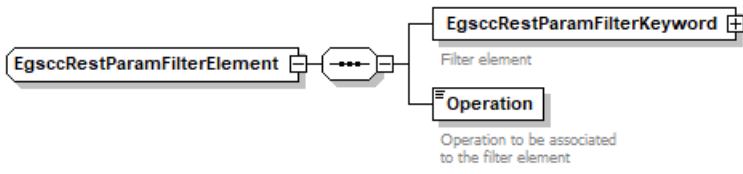
G.80 complexType EgsccrestPktTmFilterKeyword

diagram	
namespace	http://edds.egos.esa/model
children	Apid Live PacketId PacketInstanceld PusSubType PusType QualityFlag SourceSequenceCounter SpacecraftId VirtualChannelId ReceptionRoute
used by	element EgsccrestTmFilterElement/EgsccrestPktTmFilterKeyword

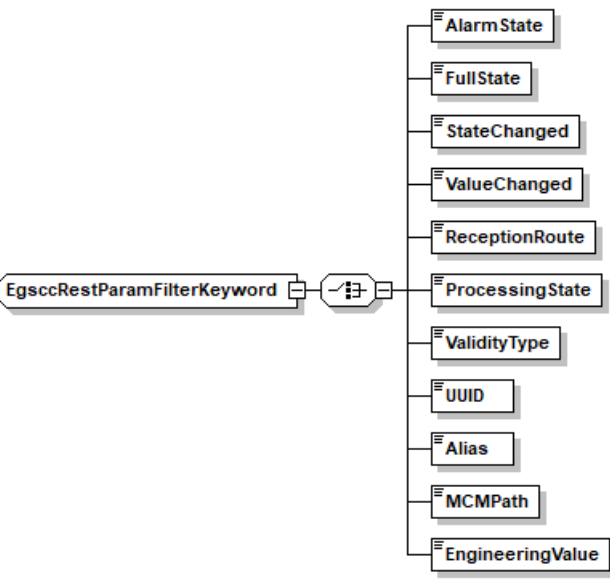
G.81 complexType EgscRestParamFilterList

diagram	
namespace	http://edds.egos.esa/model
children	EgscRestParamFilterElement
used by	element EgscRestParamTmFilter/EgscRestParamFilterList

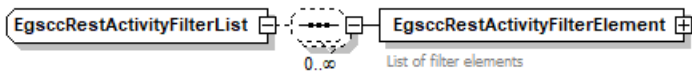
G.82 complexType EgscRestParamFilterElement

diagram	
namespace	http://edds.egos.esa/model
children	EgscRestPktTmFilterKeyword Operation
used by	element EgscRestTmFilterList/EgscRestTmFilterListElement

G.83 complexType EgscRestParamFilterKeyword

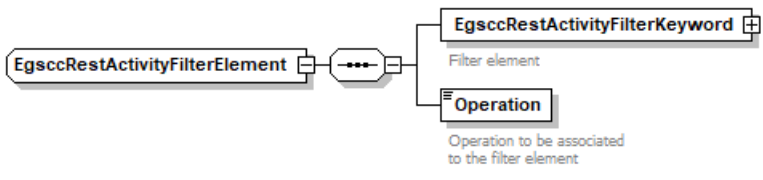
diagram	
namespace	http://edds.egos.esa/model
children	AlarmState FullState StateChanged ValueChanged ReceptionRoute ProcessingState ValidityType UUID Alias MCMPPath EngineeringValue
used by	element EgscRestParamFilterElement/EgscRestParamFilterKeyword

G.84 complexType EgscRestActivityFilterList

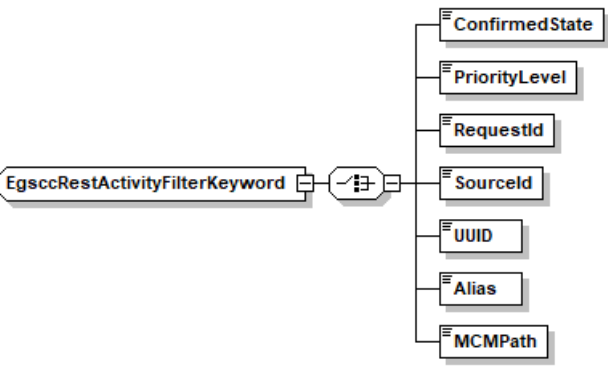
diagram	
namespace	http://edds.egos.esa/model

children	EgscRestActivityFilterElement	
used by	element	EgscRestActivityFilter/EgscRestActivityFilterList

G.85 complexType EgscRestActivityFilterElement

diagram		
namespace	http://edds.egos.esa/model	
children	EgscRestActivityFilterKeyword Operation	
used by	element	EgscRestActivityFilterList/EgscRestActivityFilterElement

G.86 complexType EgscRestActivityFilterKeyword

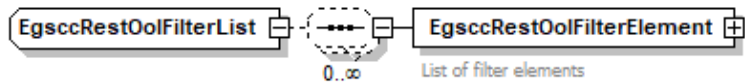
diagram		
namespace	http://edds.egos.esa/model	
children	ConfirmedState PriorityLevel RequestId SourceId UUID Alias MCMPPath	
used by	element	EgscRestActivityFilterElement/EgscRestActivityFilterKeyword

G.87 simpleType EgscTimeFiltering

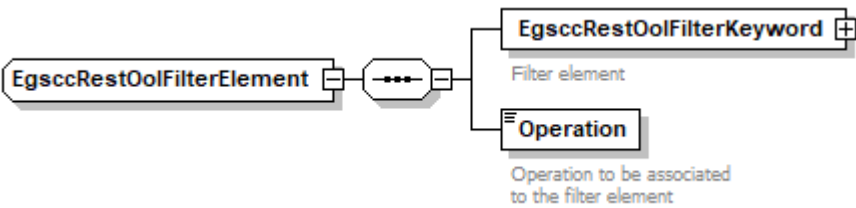
namespac e	http://edds.egos.esa/model	
type	restriction of xs:string	
properties	base	xs:string
used by	elements	EgscRestTmPacketFilter/EgscTimeFiltering EgscRestTcPacketFilter/EgscTimeFiltering EgscRestLogMessageFilter/EgscTimeFiltering EgscRestActivityFilter/EgscTimeFiltering EgscRestParamTmFilter/EgscTimeFiltering

facets	Kind	Value	annotation
	enumeration	GENERATION_TIME	documentation On board Time of the packet (Primary Key)
	enumeration	RECEPTION_TIME	documentation Reception time
	enumeration	PROCESSING_TIME	documentation Processing time
	enumeration	STORAGE_TIME	documentation Storage time
	enumeration	EXECUTION_TIME	documentation Execution time

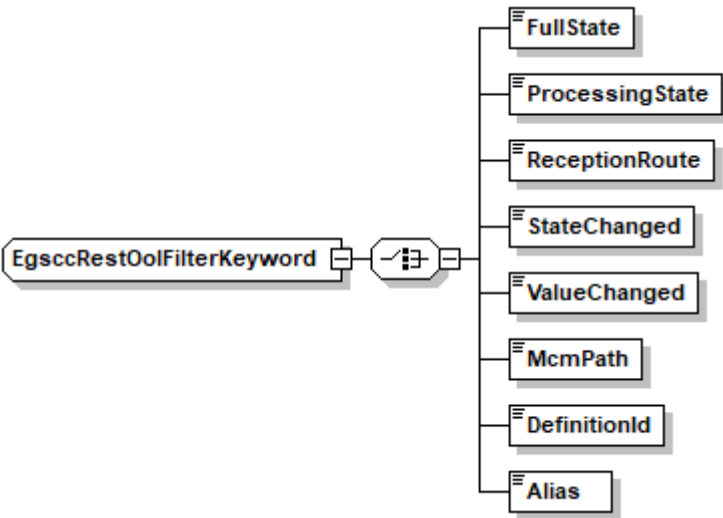
G.88 complexType EgscRestOolFilterList

diagram	
namespace	http://edds.egos.esa/model
children	EgscRestOolFilterElement
used by	element EgscRestOolFilter/EgscRestOolFilterList

G.89 complexType EgscRestOolFilterElement

diagram	
namespace	http://edds.egos.esa/model
children	EgscRestOolFilterKeyword Operation
used by	element EgscRestOolFilterList/EgscRestOolFilterElement

G.90 complexType EgscRestOolFilterKeyword

diagram	
---------	--

namespace	http://edds.egos.esa/model
children	FullState ProcessingState ReceptionRoute StateChanged ValueChanged McmPath DefinitionId Alias
used by	element EgscclRestOolFilterElement/EgscclRestOolFilterKeyword

Appendix H Format Data Type

H.1 group *SplittableFormat*

diagram	Defines the base properties for splittable formats.
namespace	http://edds.egos.esa/model
children	SplitFileSize
annotation	documentation Defines the base properties for splittable formats.

This section of the document describes the format types as defined in the format.xsd file.

H.2 complexType *CatalogueFormatting*

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.3 complexType *EddsUsageReportFormatting*

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.4 complexType EvPacketFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.

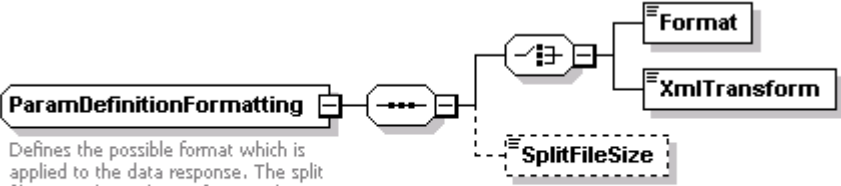
H.5 complexType PacketFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.

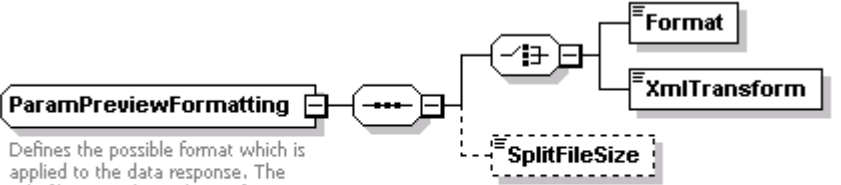
H.6 complexType PacketStatisticsFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

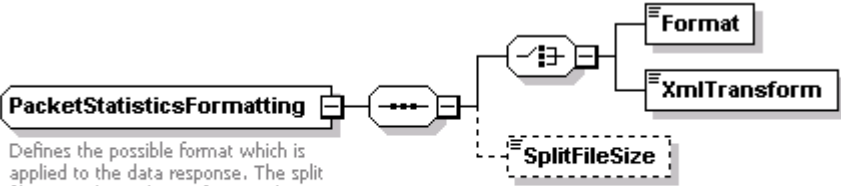
H.7 complexType ParamDefinitionFormatting

diagram	 Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.8 complexType ParamPreviewFormatting

diagram	 Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.9 complexType ParamStatisticsFormatting

diagram	 Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.10 complexType ParamTmFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.11 complexType ParcRawPacketFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.12 complexType PktTmGapReportFormat

diagram	The possible formatting options for file system catalogue response
namespace	http://edds.egos.esa/model
children	XML ASCII BINARY
annotation	documentation The possible formatting options for file system catalogue response

H.13 complexType ReportFormatting

diagram	The diagram shows the structure of the ReportFormatting complexType. It starts with a root box labeled ReportFormatting. This box is connected to a sequence container (a rectangle with three dots). The sequence container is connected to a choice container (a rectangle with a vertical bar and a plus sign). The choice container has two branches: one leading to a box labeled Format and another leading to a box labeled XmlTransform. A dashed line also connects the sequence container to a dashed box labeled SplitFileSize. Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.14 complexType SMONParamFormatting

diagram	The diagram shows the structure of the SMONParamFormatting complexType. It starts with a root box labeled SMONParamFormatting. This box is connected to a sequence container (a rectangle with three dots). The sequence container is connected to a choice container (a rectangle with a vertical bar and a plus sign). The choice container has two branches: one leading to a box labeled Format and another leading to a box labeled XmlTransform. A dashed line also connects the sequence container to a dashed box labeled SplitFileSize. Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response.

H.15 complexType EgsgccActivityFormatting

diagram	The diagram shows the structure of the EgsgccActivityFormatting complexType. It starts with a root box labeled EgsgccActivityFormatting. This box is connected to a sequence container (a rectangle with three dots). The sequence container is connected to a choice container (a rectangle with a vertical bar and a plus sign). The choice container has two branches: one leading to a box labeled Format and another leading to a box labeled XmlTransform. A dashed line also connects the sequence container to a dashed box labeled SplitFileSize. Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
annotation	documentation Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken..

H.16 simpleType ArchiveFileFormat

namespace	http://edds.egos.esa/model
type	restriction of xs:string
properties	base xs:string
facets	Kind Value annotation enumeration BINARY
annotation	documentation Defines the list of supported format of the data response.

H.17 simpleType CatalogueFormat

namespace	http://edds.egos.esa/model
type	restriction of xs:string
properties	base xs:string
used by	element CatalogueFormatting/Format
facets	Kind Value annotation enumeration XML enumeration ASCII
annotation	documentation Defines the list of supported format of the data response.

H.18 simpleType EddsUsageReportFormat

namespace	http://edds.egos.esa/model
type	restriction of xs:string
properties	base xs:string
used by	element EddsUsageReportFormatting/Format
facets	Kind Value annotation enumeration XML
annotation	documentation Defines the list of supported format of the data response.

H.19 simpleType EvPacketFormat

namespace	http://edds.egos.esa/model
type	restriction of xs:string
properties	base xs:string
facets	Kind Value annotation enumeration EDDS_RAW
annotation	documentation Defines the list of supported format of the data response.

H.20 simpleType PacketFormat

namespace	http://edds.egos.esa/model
-----------	----------------------------

type	restriction of xs:string		
properties	base	xs:string	
facets	Kind	Value	annotation
	enumeration	EDDS_RAW	
	enumeration	SFDU	
	enumeration	GDDS_BINARY	
annotation	documentation Defines the list of supported format of the data response.		

H.21 simpleType PacketStatisticsFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	PacketStatisticsFormatting/Format	
facets	Kind	Value	annotation
	enumeration	XML	
annotation	documentation Defines the list of supported format of the data response.		

H.22 simpleType ParamDefinitionFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ParamDefinitionFormatting/Format	
facets	Kind	Value	annotation
	enumeration	XML	
	enumeration	BINARY	
annotation	documentation Defines the list of supported format of the data response.		

H.23 simpleType ParamPreviewFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ParamPreviewFormatting/Format	
facets	Kind	Value	annotation
	enumeration	XML	
	enumeration	ASCII	
annotation	documentation Defines the list of supported format of the data response.		

H.24 *simpleType ParamStatisticsFormat*

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ParamStatisticsFormatting/Format	
facets	Kind	Value	annotation
	enumeration	XML	
	enumeration	ASCII	
annotation	documentation Defines the list of supported format of the data response.		

H.25 *simpleType ParamTmFormat*

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ParamTmFormatting/Format	
facets	Kind	Value	annotation
	enumeration	XML	documentation XML format
	enumeration	TDRS	documentation TDRS spreadsheet format
	enumeration	Binary	documentation EGS-CC Binary format
	enumeration	DARC_Binary	documentation DARC Binary format
annotation	documentation Defines the list of supported format of the data response.		

H.26 *simpleType ParcRawPacketFormat*

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ParcRawPacketFormatting/Format	
facets	Kind	Value	annotation
	enumeration	XML	
	enumeration	BINARY	
annotation	documentation Defines the list of supported format of the data response.		

H.27 *simpleType ReportFormat*

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	ReportFormatting/Format	

facets	Kind	Value	annotation
	enumeration	XML	
	enumeration	ASCII	
	enumeration	BINARY	
annotation	documentation Defines the list of supported format of the data response.		

H.28 simpleType SMONParamFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
used by	element SMONParamFormatting/Format		
facets	Kind	Value	annotation
	enumeration	XML	documentation XML format
	enumeration	DARC_XML	documentation DARC XML format
	enumeration	DARC_BINARY	documentation DARC binary format
	enumeration	TDRS	documentation TDRS spreadsheet format
	enumeration	BINARY	documentation Binary format
annotation	documentation Defines the list of supported format of the data response.		

H.29 simpleType EgscclActivityFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
used by	element EgscclActivityFormatting		
facets	Kind	Value	annotation
	enumeration	XML	documentation XML format
	enumeration	BINARY	documentation Binary format
annotation	documentation Defines the list of supported format of the data response.		

H.30 complexType EgscclLogMessageFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFile Size

used by	element EgscclLogMessageReport/EgscclLogMessageFormat
annotation	documentation Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken..

H.31 simpleType EgscclLogMessageFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
used by	element EgscclLogMessageFormatting/Format		
facets	Kind	Value	annotation
	enumeration	XML	documentation XML format
	enumeration	BINARY	documentation Binary format
annotation	documentation Defines the list of supported format of the data response.		

H.32 complexType EgscclRestLogMessageFormatting

diagram	UML diagram of the complex type EgscclRestLogMessageFormatting. The diagram shows a sequence of elements: EgscclRestLogMessageFormatti... (with documentation: "Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken."), followed by an optional element (indicated by a dashed line), then a choice between Format and XmlTransform, and finally an optional element SplitFile Size (indicated by a dashed line).
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFileSize
used by	element EgscclRestLogMessageReport/EgscclRestLogMessageFormat
annotation	documentation Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken

H.33 simpleType EgscclRestLogMessageFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base xs:string		
used by	element EgscclRestLogMessageFormatting/Format		
facets	Kind	Value	annotation
	enumeration	XML	documentation XML format
	enumeration	BINARY	documentation Binary format
annotation	documentation Defines the list of supported format of the data response.		

H.34 simpleType EgscRestParamFormat

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	EgscRestParamFormatting/Format	
facets	Kind	Value	annotation
	enumeration	Binary	documentation EGS-CC Binary format
	enumeration	XML	documentation XML format
	enumeration	TDRS	documentation TDRS spreadsheet format
	enumeration	DARC_Binary	documentation DARC Binary format
annotation	documentation Defines the list of supported format of the data response.		

H.35 complexType EgscRestParamFormatting

diagram	Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.
namespace	http://edds.egos.esa/model
children	Format XmlTransform SplitFile Size
used by	element EgscRestParam/EgscRestParamFormat
annotation	documentation Defines the possible format which is applied to the data response. The split file size only applies to formats that can be split and is in megabytes. If omitted, the default from EDDS Server is taken.

Appendix I Report Data Type

This section of the document describes the report complex types as defined in the report.xsd file.

I.1 complexType ActivityDetails

diagram	
namespace	http://edds.egos.esa/model
children	Name DefinitionID Alias Path Description
used by	EgscActivityReportListElement/ActivityDetails

I.2 complexType ActivityInvokeDetails

diagram	
namespace	http://edds.egos.esa/model
children	Name Path Description ProcessingTime GenerationTime GroundScheduleTime ExternalExecutionTime EarliestExecutionStartTime LatestExecutionEndTime ActivityState MonitoringUpdateType Suspended RouteId SourceId AssumedState OccurrenceId ServiceAccessPoint RequestId ActivityArgumentData VerificationStage
used by	EgscActivityReportListElement/ActivityInvokeDetails

I.3 complexType DataRequestBase

diagram	<pre>classDiagram class DataRequestBase { +DataSource +RequestPostProcessing +Delivery } class DataSource { +Defines the data source associated with the request } class RequestPostProcessing { +Defines if the request response has to be compressed or and encrypted. } class Delivery { +Defines the mechanism used to deliver the data response to the user. } DataRequestBase --> DataSource DataRequestBase --> RequestPostProcessing DataRequestBase --> Delivery</pre>
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery
used by	complexType EddsUsageReport EventRecordReport OolRecordReport PktTcReport PktTmReport
annotation	documentation Base class for all batch requests (including report requests)

1.4 complexType EddsUsageReport

diagram	EddsUsageReport Defines the Edds Usage report DataRequestBase (extension) DataSource Defines the data source associated with the request RequestPostProcessing Defines if the request response has to be compressed or and encrypted. Delivery Defines the mechanism used to deliver the data response to the user. EddsUsageReportFilter Defines the Edds Usage Report Filter EddsUsageReportFormat Defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery EddsUsageReportFilter EddsUsageReportFormat
annotation	documentation Defines the Edds Usage report

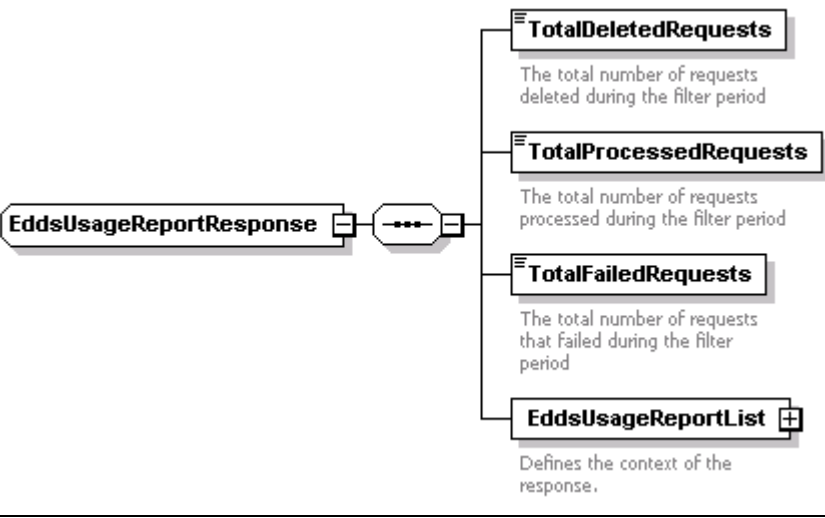
1.5 complexType EddsUsageReportList

diagram	EddsUsageReportList A list of Edds usage elements EddsUsageReportListElement
namespace	http://edds.egos.esa/model
children	EddsUsageReportListElement
used by	element EddsUsageReportResponse/EddsUsageReportList
annotation	documentation A list of Edds usage elements

1.6 complexType EddsUsageReportListElement

diagram	The diagram illustrates the structure of the EddsUsageReportListElement complex type. It consists of a central box labeled EddsUsageReportListElement connected to a vertical stack of attribute boxes. The attributes are: RequestId: The ID of the request User: The user who submitted the request Role: Role of the user who submitted the request Status: Role of the user who submitted the request ReasonForFailure: The reason why the request failed SizeOfResponseFile: The size of response file SubmissionDate: The date the request was created/submitted ExecutionDate: The date the request was/is planned to be executed CompletionDate: The date the request was completed
namespace	http://edds.egos.esa/model
children	RequestId User Role Status ReasonForFailure SizeOfResponseFile SubmissionDate ExecutionDate CompletionDate
used by	element EddsUsageReportList/EddsUsageReportListElement

I.7 *complexType EddsUsageReportResponse*

diagram	
namespace	http://edds.egos.esa/model
children	TotalDeletedRequests TotalProcessedRequests TotalFailedRequests EddsUsageReportList

I.8 *complexType EgscclActivityReport*

diagram	
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgscclActivityFilter EgscclActivityFormat

I.9 *complexType EgscclActivityReportList*

diagram	
namespace	http://edds.egos.esa/model
children	EgscclActivityReportListElement
used by	EgscclActivityReportResponse/EgscclActivityReportList

I.10 *complexType EgscclActivityReportListElement*

diagram	
namespace	http://edds.egos.esa/model
children	ActivityInvokeDetails ActivityDetails
used by	EgscclActivityReportList/EgscclActivityReportListElement

I.11 *complexType EgscclActivityReportResponse*

diagram	
namespace	http://edds.egos.esa/model
children	EgscclActivityReportList

I.12 *complexType EgscclPktTcReport*

diagram	
---------	--

namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgscdTcPacketFilter ReportFormat

I.13 complexType EgscdTcReportList

diagram	
namespace	http://edds.egos.esa/model
children	EgscdTcReportListElement
used by	EgscdTcReportResponse/EgscdTcReportList

I.14 complexType EgscdTcReportListElement

diagram	
namespace	http://edds.egos.esa/model
children	Apid Pid PinstanceId GenerationTime EarthReceivedTime EgscdReceptionTime Live Type SubType QualityFlag TimeQuality SourceSequenceCounter SpacecraftId VirtualChannelId ActivityRoute ActivityOccurrence Annotations RawBodyData DataPartition StorageTime TransmissionMode
used by	EgscdTcReportList/EgscdTcReportListElement

I.15 complexType EgscdTcReportResponse

diagram	
namespace	http://edds.egos.esa/model
children	EgscdTcReportList

I.16 complexType EgscdTmReport

diagram	
namespace	http://edds.egos.esa/model
children	DataSource RequestPostProcessing Delivery EgscdTmPacketFilter ReportFormat

I.17 complexType EgscdTmReportList

diagram	
namespace	http://edds.egos.esa/model
children	EgscdTmReportListElement
used by	EgscdTmReportResponse/EgscdTmReportList

I.18 complexType EgscdTmReportListElement

diagram	
namespace	http://edds.egos.esa/model
children	Apid Pid PinstanceId GenerationTime EarthReceivedTime EgscdReceptionTime Live Type SubType QualityFlag TimeQuality SourceSequenceCounter SpacecraftId VirtualChannelId RawBodyData Datapartition StorageTime Annotations TransactionId ReceptionRoute ProviderId AntennaId IdleFrame MasterChannelFrameCounter MasterChannelId VirtualChannelFrameCounter DataLinkContinuity
used by	EgscdTmReportList/EgscdTmReportListElement

I.19 complexType EgscPktTmReportResponse

diagram	
namespace	http://edds.egos.esa/model
children	EgscPktTmReportList

I.20 complexType EventRecordReport

diagram	
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery EvPacketFilter ReportFormat
annotation	documentation Defines filter to be applied for the Ev report

I.21 complexType EventRecordReportList

diagram	
namespace	http://edds.egos.esa/model
children	EventRecordReportListElement
used by	element EventRecordReportResponse/EventRecordReportList
annotation	documentation A list of EV packets

I.22 complexType EventRecordReportListElement

diagram	The diagram shows the structure of the EventRecordReportListElement complex type. It is a container element (rectangle with a small square on the left) that contains a sequence of elements (indicated by a dashed line with three dots). The elements are: Id (The name of the event), Category (The category of the event), Domain (The domain of the event), EventId (The event ID of the event), EventType (The event type of the event), GenerationTime (The generation time of the event), Source (The source of the event), Application (The application of the event), Severity (The severity of the event), Type (The type of the event), Workstation (The workstation of the event), and Message (The message of the event). Each element is represented by a rectangle with a small square on the left and a description below it.
namespace	http://edds.egos.esa/model
children	Id Category Domain EventId EventType GenerationTime Source Application Severity Type Workstation Message
used by	element EventRecordReportList/EventRecordReportListElement

I.23 complexType EventRecordReportResponse

diagram	The diagram shows the structure of the EventRecordReportResponse complex type. It is a container element (rectangle with a small square on the left) that contains a sequence of elements (indicated by a dashed line with three dots). The elements are: EventRecordReportList (Defines the context of the request.). Each element is represented by a rectangle with a small square on the left and a description below it.
namespace	http://edds.egos.esa/model

children	EventRecordReportList
----------	-----------------------

I.24 complexType OolDataReportList

diagram	Defines the list of OOL Data Report responses 0..∞ Defines the OOL Data Report response element
namespace	http://edds.egos.esa/model
children	OolDataReportResponseElement
used by	element OolDataReportResponse/OolDataReportList
annotation	documentation Defines the list of OOL Data Report responses

I.25 complexType OolDataReportResponse

diagram	Defines the context of the request.
namespace	http://edds.egos.esa/model
children	OolDataReportList

1.26 complexType OolDataReportResponseElement

diagram	<pre> classDiagram class OolDataReportResponseElement { RecordType GenerationTime ParameterChangedTime ParameterName ParameterState OolState Value OriginalValue LowerLimit UpperLimit AllowedValues Scclimit ReferenceValue DeltaValue MinValue LastValue ViolationNumber ViolationNumberHard ViolationNumberSoft Description Unit BehaviourLimitState BehaviourDeltaState BehaviourScclState BehaviourStatusState ViolationDescription } </pre> OolDataReportResponseElement Defines the OOL Data Report response element RecordType: One of LIMIT, STATE, STATUS_CONSISTENCY, SOFT_HARD_LIMIT, LIMIT_MO, DELTA GenerationTime: Generation time stamp as CCSDS AscIIA (yyyy-mm-ddThh:mm:ss.mmm) ParameterChangedTime: Parameter changed time stamp as CCSDS AscIIA (yyyy-mm-ddThh:mm:ss.mmm) ParameterName ParameterState: One of NOMINAL, WARNING, OUT_OF_LIMITS, SCC, SCC_DISABLE, SCC_OFF, SCC_UNINIT, VIOLATION OolState Value OriginalValue LowerLimit: Only for LIMIT, SOFT_HARD_LIMIT, LIMIT_MO types UpperLimit: Only for LIMIT, SOFT_HARD_LIMIT, LIMIT_MO types AllowedValues: Each value should be comma separated Scclimit: Only for STATUS_CONSISTENCY types ReferenceValue: Only for DELTA types DeltaValue: Only for DELTA types MinValue: Only for DELTA types LastValue: Only for DELTA types from data decoded from the PARC ViolationNumber: Not for DELTA types ViolationNumberHard: Only for SOFT_HARD_LIMIT types ViolationNumberSoft: Only for SOFT_HARD_LIMIT types Description Unit BehaviourLimitState: Only for the SOFT_HARD_LIMIT and LIMIT types from the data provision services BehaviourDeltaState: Only for the DELTA type from the data provision services BehaviourScclState: Only for the STATUS_CONSISTENCY type from the data provision services BehaviourStatusState: Only for the STATE type from the data provision services ViolationDescription: Only for the LIMIT_MO types from the data provision services
namespace	http://edds.egos.esa/model
children	RecordType GenerationTime ParameterChangedTime ParameterName ParameterState OolState Value OriginalValue LowerLimit UpperLimit AllowedValues Scclimit ReferenceValue DeltaValue MinValue LastValue ViolationNumber ViolationNumberHard ViolationNumberSoft Description Unit BehaviourLimitState BehaviourDeltaState BehaviourScclState BehaviourStatusState ViolationDescription

	LastValue ViolationNumber ViolationNumberHard ViolationNumberSoft Description Unit BehaviourLimitState BehaviourDeltaState BehaviourSccState BehaviourStatusState ViolationDescription
used by	element OolDataReportList/OolDataReportResponseElement

1.27 complexType OolRecordReport

diagram	<pre>graph LR OolRecordReport[OolRecordReport Defines the Ool data report] subgraph DataRequestBase_extension [DataRequestBase (extension)] DataSource[DataSource Defines the data source associated with the request] RequestPostProcessing[RequestPostProcessing Defines if the request response has to be compressed or and encrypted.] Delivery[Delivery Defines the mechanism used to deliver the data response to the user.] end OolDataReportFilter[OolDataReportFilter Defines the Ool Data Report Filter] ReportFormat[ReportFormat Defines the possible format which is applied to the data response.] OolRecordReport --- Seq1[...] Seq1 --- DataRequestBase_extension Seq1 --- Seq2[...] Seq2 --- OolDataReportFilter Seq2 --- ReportFormat</pre>
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery OolDataReportFilter ReportFormat
annotation	documentation Defines the Ool data report

I.28 complexType PktTcReport

diagram	The diagram shows the structure of the PktTcReport complex type. It is an extension of DataRequestBase. The PktTcReport type contains a DataSource element, a RequestPostProcessing element, and a Delivery element. Additionally, it contains a TcPacketFilter element and a ReportFormat element. The DataSource element defines the data source associated with the request. The RequestPostProcessing element defines if the request response has to be compressed or and encrypted. The Delivery element defines the mechanism used to deliver the data response to the user. The TcPacketFilter element defines the basic information necessary to retrieve the data from the Report and defines the filter to be applied to the retrieved data. The ReportFormat element defines the possible format which is applied to the data response.
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery TcPacketFilter ReportFormat
annotation	documentation Defines filter to be applied for the TC report

I.29 complexType PktTcReportList

diagram	The diagram shows the structure of the PktTcReportList complex type. It is a list of PktTcReportListElement elements. The multiplicity is 0..∞.
namespace	http://edds.egos.esa/model
children	PktTcReportListElement
used by	element PktTcReportResponse/PktTcReportList
annotation	Documentation A list of TC packets

1.30 complexType PktTcReportListElement

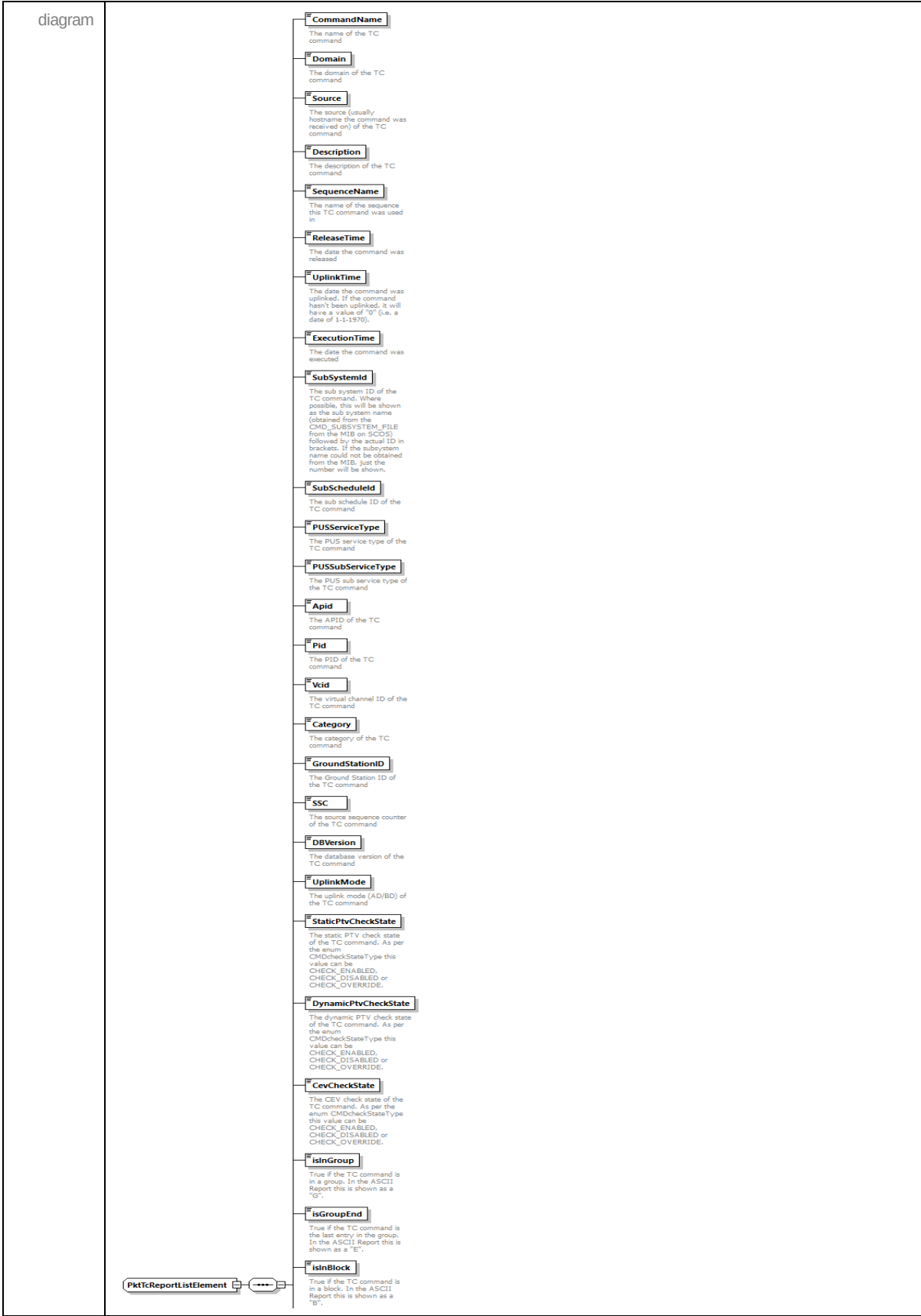


diagram
continued

isBlockEnd
True if the TC command is the last entry in the block. In the ASCII Report this is shown as a 'E'.

InterlockType
The interlock type of the TC command. The mapping between the ASCII report is as follows:
NONE, IL_LOCAL, IL_GLOBAL, IL_SUBSYS_LOCAL, IL_SUBSYS_GLOBAL, SUBSYS_GLOBAL, IL_FAIL_ONLY_LOCAL, IL_FAIL_ONLY_SUBSYS, T

InterlockStageType
The interlock stage type of the TC command. The mapping between the ASCII report is as follows:
MCS_RELEASE - M, UV_GS_RECVIE - R, UV_GS_UPLINK - U, UV_ONB_ACCEPT - O, EV_APP_ACCEPT - A, EV_START_EXEC - S, EV_END_EXEC - C, EV_PROGRESS_0 - 0 (continues up to EV_PROGRESS_9 - 9)

SrcType
The source type of the TC command

FrameCount
The frame count of the TC command. "-" is shown if uplink mode is BD

TransmissionCount
The transmission count of the TC command.

ReleaseState
The release state of the TC command

GroundState
The ground state of the TC command

UplinkState
The uplink state of the TC command

OnBoardState
The on-board state of the TC command

OnBoardAccState
The on-board acceptance state of the TC command

OnBoardAccPBState
The on-board acceptance state of the TC command based on Play Back TM

ExecStartRTState
Execution start state for Real Time TM

ExecStartPBState
Execution start state for Play Back TM

ExecStep1RTState
Execution step 1 state for Real Time TM

ExecStep1PBState
Execution step 1 state for Play Back TM

ExecStep2RTState
Execution step 2 state for Real Time TM

ExecStep2PBState
Execution step 2 state for Play Back TM

ExecCompState
The execution state of the TC command

ExecCompPBState
The execution Play Back state of the TC command

TtObqStatus
The time tagged on-board queue status of the TC command. (Only valid for time tagged commands).

TtDelStatus
The deletion status of the time tagged TC command. (Only valid for time tagged commands).

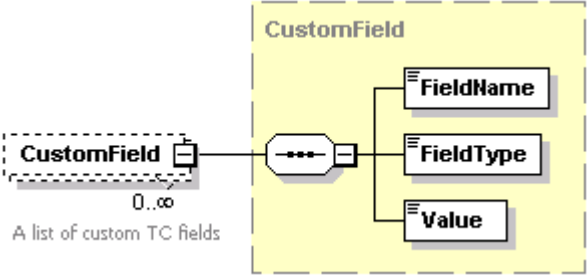
TtLoadStatus
The load status of the time tagged TC command. (Only valid for time tagged commands).

CustomField
A list of custom TC fields

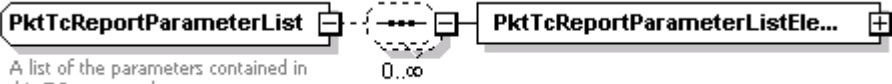
PktTcReportParameterList
The parameters contained in this

Generated by XMLSpy
www.altova.com

I.31 complexType CustomField

diagram	
type	CustomField
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	FieldName FieldType Value
annotation	documentation A list of custom TC fields

I.32 complexType PktTcReportParameterList

diagram	
namespace	http://edds.egos.esa/model
children	PktTcReportParameterListElement
used by	element PktTcReportListElement/PktTcReportParameterList
annotation	documentation A list of the parameters contained in this TC command

I.33 complexType PktTcReportParameterListElement

diagram	The diagram shows the structure of the PktTcReportParameterListElement complex type. It consists of a sequence of elements: ParameterName, ParameterDescription, ParameterUnit, ParamType, ParamRep, FixEdit, HasChanged, Radix, IsEditable, ManuallyEdited, and ParamValue. Each element is represented by a box with a title and a description. The PktTcReportParameterListElement box is connected to the FixEdit box by a line with a circle and three dots, indicating a sequence relationship.
namespace	http://edds.egos.esa/model
children	ParameterName ParameterDescription ParameterUnit ParamType ParamRep FixEdit HasChanged Radix IsEditable ManuallyEdited ParamValue
used by	element PktTcReportParameterList/PktTcReportParameterListElement

I.34 complexType PktTcReportResponse

diagram	The diagram shows the structure of the PktTcReportResponse complex type. It consists of a sequence of elements: PktTcReportResponse and PktTcReportList. The PktTcReportResponse box is connected to the PktTcReportList box by a line with a circle and three dots, indicating a sequence relationship. The PktTcReportList box has a description: "Defines the context of the request."
namespace	http://edds.egos.esa/model

children	PktTcReportList
----------	-----------------

I.35 complexType PktTmReport

diagram	
namespace	http://edds.egos.esa/model
type	extension of DataRequestBase
properties	base DataRequestBase
children	DataSource RequestPostProcessing Delivery TmPacketFilter ReportFormat
annotation	documentation Defines filter to be applied for the TM report

I.36 complexType PktTmReportList

diagram	
namespace	http://edds.egos.esa/model
children	PktTmReportListElement
used by	element PktTmReportResponse/PktTmReportList
annotation	documentation A list of TM packets

I.37 complexType PktTmReportListElement

diagram	
namespace	http://edds.egos.esa/model
children	Apid Pid Category DatabaseVersion GenerationTime DataStream PartitionName Domain GroundStation QualityFlag ReceptionTime SleServiceId SourceSequenceCounter Spid Type SubType TimeQuality

	VirtualChannel DataUnitType Pi1 Pi2 PacketLength Mnemonic Description SpacecraftId Id PktTmReportParameterList RawBodyData
used by	element PktTmReportList/PktTmReportListElement

I.38 complexType PktTmReportParameterList

diagram	
namespace	http://edds.egos.esa/model
children	PktTmReportParameterListElement
used by	element PktTmReportListElement/PktTmReportParameterList

I.39 complexType PktTmReportParameterListElement

diagram	
namespace	http://edds.egos.esa/model
children	ParameterName ParameterDescription ParameterUnit ParameterRadix FilingTime ParamValue BitOffset ByteOffset Length
used by	element PktTmReportParameterList/PktTmReportParameterListElement

I.40 complexType PktTmReportResponse

diagram	Defines the context of the request.
namespace	http://edds.egos.esa/model
children	PktTmReportList

I.41 complexType PktTmGapReportList

diagram	A list of TM Gaps 0..∞
namespace	http://edds.egos.esa/model
children	PktTmGapReportListElement
used by	element PktTmGapReportResponse/PktTmGapReportList

I.42 complexType PktTmGapReportListElement

diagram	The start gap time The end gap time The APID of the TM gap The data partition of the TM command The start source sequence counter The end source sequence counter The number of packets missing
namespace	http://edds.egos.esa/model
children	StartGapTime EndGapTime Apid DataPartition StartSscPkt EndSscPkt NumPktsMissing
used by	element PktTmGapReportList/PktTmGapReportListElement

I.43 *complexType* *PktTmGapReportResponse*

diagram	PktTmGapReportResponse ... PktTmGapReportList Defines the context of the request.
namespace	http://edds.egos.esa/model
children	PktTmGapReportList

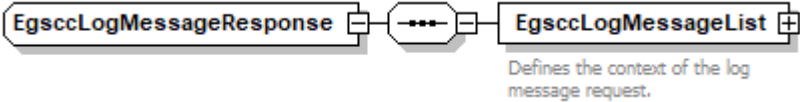
I.44 complexType EgsccLogMessageListElement

diagram	
namespace	http://edds.egos.esa/model
children	Id LogbookId NodeId GenerationTime EgsccReceptionTime StorageTime CategoryId ApplicationId SystemId text TraceScope TraceLevel Annotation ContextId SystemSessionId RequiresAcknowledgement CreatedBy OriginName OriginAddress AcknowledgementDetailsElement UserIdElement Severity MessageType
used by	element EgsccLogMessageList/EgsccLogMessageListElement

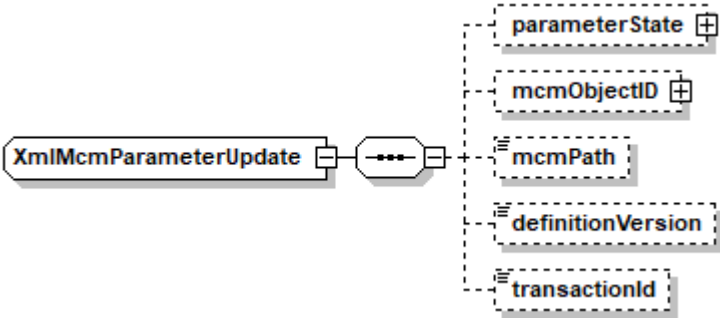
I.45 complexType EgscclLogMessageList

diagram	
namespace	http://edds.egos.esa/model
children	EgscclLogMessageListElement
used by	EgscclLogMessageResponse/EgscclLogMessageList

I.46 complexType EgscclLogMessageResponse

diagram	
namespace	http://edds.egos.esa/model
children	EgscclLogMessageList

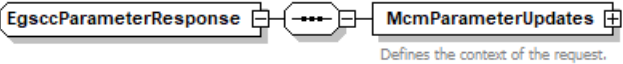
I.47 complexType XmlMcmParameterUpdate

diagram	
namespace	http://edds.egos.esa/model
children	parameterState mcmObjectID mcmPath definitionVersion transactionId
used by	element McmParameterUpdates/XmlMcmParameterUpdate

I.48 complexType McmParameterUpdates

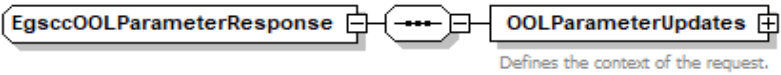
diagram	
namespace	http://edds.egos.esa/model
children	XmlMcmParameterUpdate
used by	element EgscclParameterResponse/McmParameterUpdates

I.49 complexType EgscclParameterResponse

diagram	
---------	--

namespace	http://edds.egos.esa/model
children	McmParameterUpdates

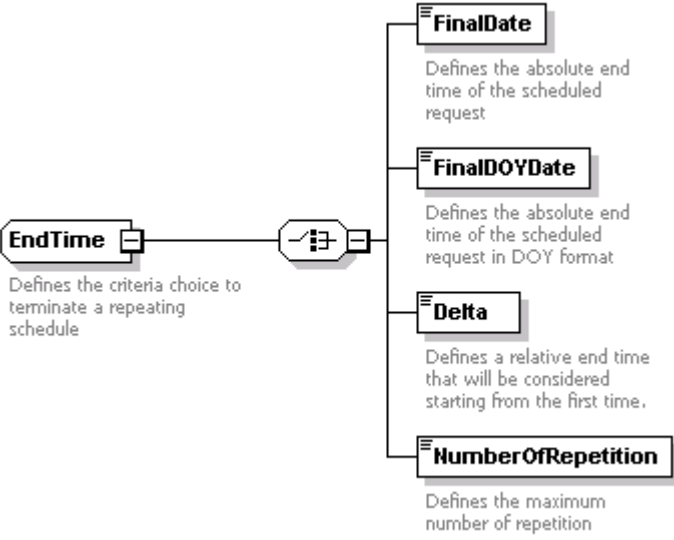
1.50 complexType EgsccOOLParameterResponse

diagram	 The diagram shows a complex type EgsccOOLParameterResponse containing an element OOLParameterUpdates. The element is optional, indicated by a circle with three dots. A note below the element states: "Defines the context of the request."
namespace	http://edds.egos.esa/model
children	OOLParameterUpdates

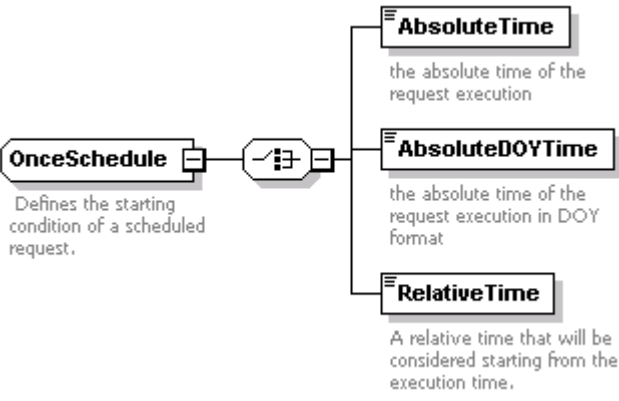
Appendix J Schedule Data Type

This section of the document describes the complex types used for scheduling as defined in the schedule.xsd file.

J.1 complexType EndTime

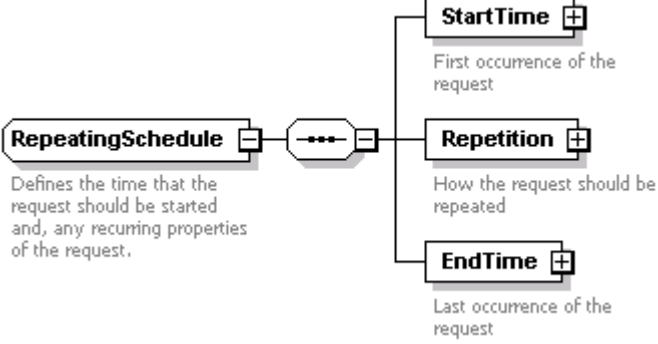
diagram	 The diagram shows the structure of the EndTime complex type. It consists of a root element EndTime (represented by a rounded rectangle with a small square on the right) containing a choice element (represented by a hexagon with a vertical line and a small square on the right). This choice element has four possible children: FinalDate, FinalDOYDate, Delta, and NumberOfRepetition. Each child is represented by a rounded rectangle with a small square on the right and a descriptive text below it. FinalDate: Defines the absolute end time of the scheduled request FinalDOYDate: Defines the absolute end time of the scheduled request in DOY format Delta: Defines a relative end time that will be considered starting from the first time. NumberOfRepetition: Defines the maximum number of repetition
namespace	http://edds.egos.esa/model
children	FinalDate FinalDOYDate Delta NumberOfRepetition
used by	element RepeatingSchedule/EndTime
annotation	documentation Defines the criteria choice to terminate a repeating schedule

J.2 complexType OnceSchedule

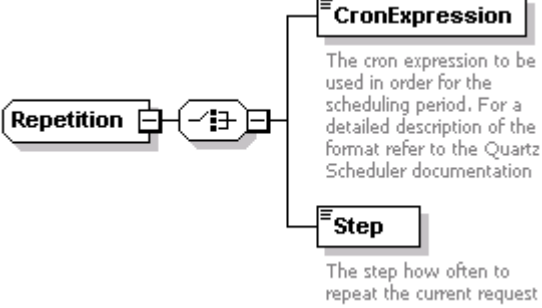
diagram	 The diagram shows the structure of the OnceSchedule complex type. It consists of a root element OnceSchedule (represented by a rounded rectangle with a small square on the right) containing a choice element (represented by a hexagon with a vertical line and a small square on the right). This choice element has three possible children: AbsoluteTime, AbsoluteDOYTime, and RelativeTime. Each child is represented by a rounded rectangle with a small square on the right and a descriptive text below it. AbsoluteTime: the absolute time of the request execution AbsoluteDOYTime: the absolute time of the request execution in DOY format RelativeTime: A relative time that will be considered starting from the execution time.
namespace	http://edds.egos.esa/model
children	AbsoluteTime AbsoluteDOYTime RelativeTime
used by	elements ScheduleOnce/Once ScheduleWithExpiry/RequestExpiryDate RepeatingSchedule/StartTime

annotation	documentation Defines the starting condition of a scheduled request.
------------	---

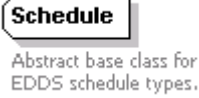
J.3 complexType RepeatingSchedule

diagram	 The diagram shows the structure of the RepeatingSchedule complexType. It consists of a RepeatingSchedule element (represented by a rounded rectangle with a small square on the left) connected to a container element (represented by a rounded rectangle with a dashed line and a small square on the left). This container element is connected to three child elements: StartTime, Repetition, and EndTime. Each child element is represented by a rounded rectangle with a small square on the left and a plus sign on the right. Below each child element is a brief description: StartTime is 'First occurrence of the request', Repetition is 'How the request should be repeated', and EndTime is 'Last occurrence of the request'. The RepeatingSchedule element has a description: 'Defines the time that the request should be started and, any recurring properties of the request.'
namespace	http://edds.egos.esa/model
children	StartTime Repetition EndTime
used by	element ScheduleRepeating/RepeatingSchedule
annotation	documentation Defines the time that the request should be started and, any recurring properties of the request.

J.4 complexType Repetition

diagram	 The diagram shows the structure of the Repetition complexType. It consists of a Repetition element (represented by a rounded rectangle with a small square on the left) connected to a container element (represented by a rounded rectangle with a dashed line and a small square on the left). This container element is connected to two child elements: CronExpression and Step. Each child element is represented by a rounded rectangle with a small square on the left and a plus sign on the right. Below each child element is a brief description: CronExpression is 'The cron expression to be used in order for the scheduling period. For a detailed description of the format refer to the Quartz Scheduler documentation', and Step is 'The step how often to repeat the current request'. The Repetition element has a description: 'Defines the time that the request should be started and, any recurring properties of the request.'
namespace	http://edds.egos.esa/model
children	CronExpression Step
used by	element RepeatingSchedule/Repetition

J.5 complexType Schedule

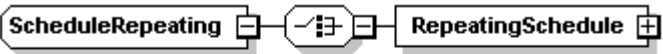
diagram	 The diagram shows the structure of the Schedule complexType. It consists of a Schedule element (represented by a rounded rectangle with a small square on the left). The Schedule element has a description: 'Abstract base class for EDDS schedule types.'
namespace	http://edds.egos.esa/model
properties	abstract true
used by	complexTypes ScheduleOnce ScheduleRepeating

annotation	documentation Abstract base class for EDDS schedule types.
------------	---

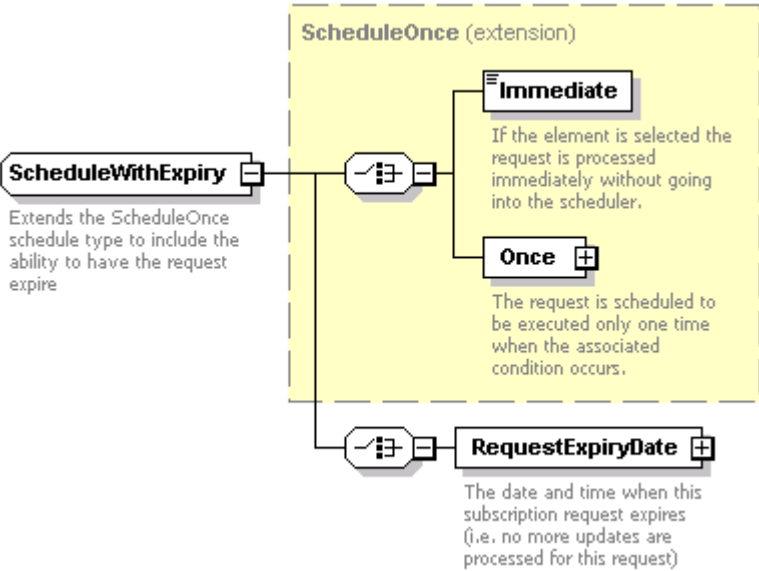
J.6 *complexType* ScheduleOnce

diagram	ScheduleOnce Defines how the request should be scheduled to run. NOTE: The XSD duration data type is used to specify a time interval. The time interval is specified in the following form "PnYnMnDnHnMnS" where: - P indicates the period (required) - nY indicates the number of years - nM indicates the number of months - nD indicates the number of days - T indicates the start of a time section (required if you are going to specify hours, minutes, or seconds) - nH indicates the number of hours - nM indicates the number of minutes - nS indicates the number of seconds The following is an example of a duration declaration in a schema: - P5Y - P5Y2M10D - P5Y2M10DT15H - PT15H Immediate If the element is selected the request is processed immediately without going into the scheduler. Once The request is scheduled to be executed only one time when the associated condition occurs.
namespace	http://edds.egos.esa/model
type	extension of Schedule
properties	base Schedule
children	Immediate Once
used by	complexType ScheduleWithExpiry
annotation	documentation Defines how the request should be scheduled to run. NOTE: The XSD duration data type is used to specify a time interval. The time interval is specified in the following form "PnYnMnDnHnMnS" where: - P indicates the period (required) - nY indicates the number of years - nM indicates the number of months - nD indicates the number of days - T indicates the start of a time section (required if you are going to specify hours, minutes, or seconds) - nH indicates the number of hours - nM indicates the number of minutes - nS indicates the number of seconds The following is an example of a duration declaration in a schema: - P5Y - P5Y2M10D - P5Y2M10DT15H - PT15H

J.7 complexType ScheduleRepeating

diagram	 With this schedule the request is repeated at defined intervals The request is scheduled to be executed more times according to the recurrence conditions.
namespace	http://edds.egos.esa/model
type	extension of Schedule
properties	base Schedule
children	RepeatingSchedule
annotation	documentation With this schedule the request is repeated at defined intervals

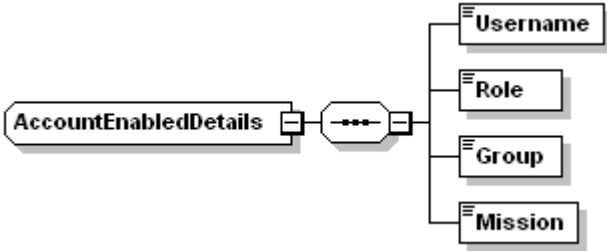
J.8 complexType ScheduleWithExpiry

diagram	 Extends the ScheduleOnce schedule type to include the ability to have the request expire ScheduleOnce (extension) Immediate If the element is selected the request is processed immediately without going into the scheduler. Once The request is scheduled to be executed only one time when the associated condition occurs. RequestExpiryDate The date and time when this subscription request expires (i.e. no more updates are processed for this request)
namespace	http://edds.egos.esa/model
type	extension of ScheduleOnce
properties	base ScheduleOnce
children	Immediate Once RequestExpiryDate
annotation	documentation Extends the ScheduleOnce schedule type to include the ability to have the request expire

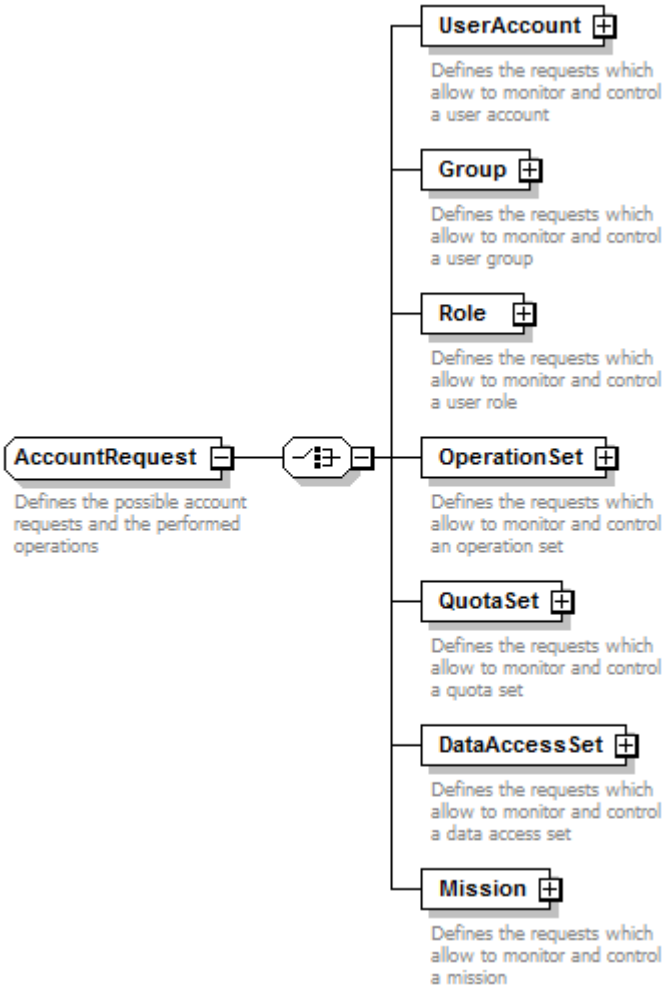
Appendix K User Management Data Type

This section of the document describes the complex types used for user mangement as defined in the userrequest.xsd file.

K.1 *complexType AccountEnabledDetails*

diagram	
namespace	http://edds.egos.esa/model
children	Username Role Group Mission
used by	element AccountEnabledDetails

K.2 *complexType AccountRequest*

diagram	
---------	--

namespace	http://edds.egos.esa/model
children	UserAccount Group Role OperationSet QuotaSet DataAccessSet Mission
used by	element AccountRequestMessagePart/AccountRequest
annotation	documentation Defines the possible account requests and the performed operations

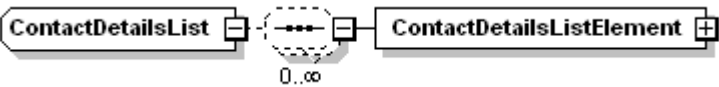
K.3 complexType AccountRequestMessagePart

diagram	
namespace	http://edds.egos.esa/model
children	AccountRequest User MissionId PrivacyTag
used by	element AccountRequestMessagePart
annotation	documentation Define the client message which describes the account management request

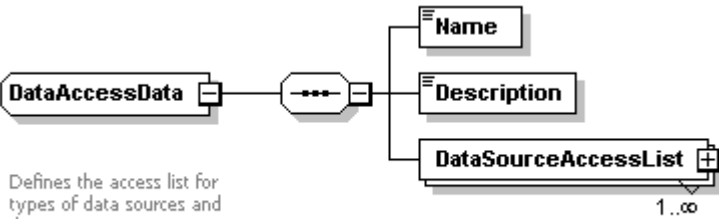
K.4 complexType ContactDetailsElement

diagram	
namespace	http://edds.egos.esa/model
children	Id Email FileServer TargetFolder UserName PostalAddress TelephoneNumber
used by	elements UserAccountBase/ContactDetailsList UserAttributes/ContactDetailsList

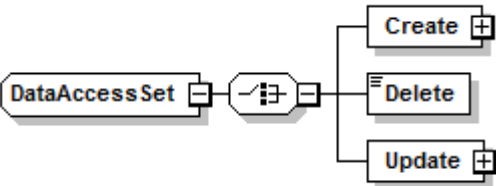
K.5 complexType ContactDetailsList

diagram	 The diagram shows a complex type ContactDetailsList containing a sequence of ContactDetailsListElement elements. The sequence is indicated by a dashed line with three dots and a cardinality of 0..∞.
namespace	http://edds.egos.esa/model
children	ContactDetailsListElement
used by	elements UserAccountData/ContactDetailsList UserAttributes/ContactDetailsList

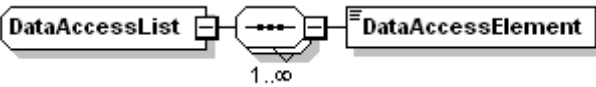
K.6 complexType DataAccessData

diagram	 The diagram shows a complex type DataAccessData containing a sequence of three elements: Name, Description, and DataSourceAccessList. The sequence is indicated by a dashed line with three dots and a cardinality of 1..∞. Below the diagram, the text reads: "Defines the access list for types of data sources and the request types."
namespace	http://edds.egos.esa/model
children	Name Description DataSourceAccessList
used by	elements DataAccessSet/Create MissionDetailsList/DataAccessDataList DataAccessSet/Update
annotation	documentation Defines the access list for types of data sources and the request types.

K.7 complexType DataAccessSet

diagram	 The diagram shows a complex type DataAccessSet containing a sequence of three elements: Create, Delete, and Update. The sequence is indicated by a dashed line with three dots and a cardinality of 1..∞.
namespace	http://edds.egos.esa/model
children	Create Delete Update
used by	element AccountRequest/DataAccessSet

K.8 complexType DataAccessList

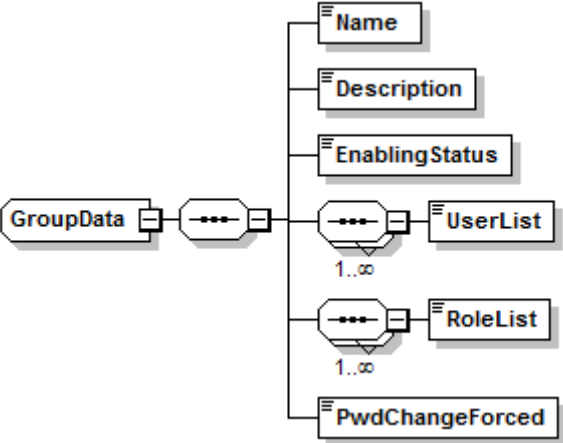
diagram	 The diagram shows a complex type DataAccessList containing a sequence of DataAccessElement elements. The sequence is indicated by a dashed line with three dots and a cardinality of 1..∞.
namespace	http://edds.egos.esa/model
children	DataAccessElement

used by	element DataAccessData/DataAccessList
---------	--

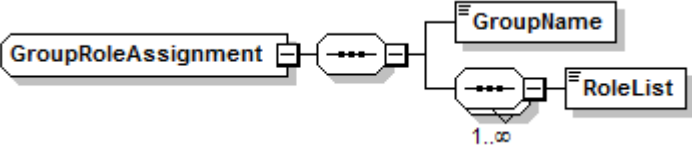
K.9 complexType Group

diagram	The diagram illustrates the structure of the Group complexType. It features a central Group element box with the description: "Defines the operations which can be performed to manipulate a user group". This box is connected to a container box (a rectangle with a dashed border and a small square icon), which then branches out to a vertical list of seven operation boxes. Each operation box has a title and a description: Create (+): Allows creating an new user group Delete (=): Allows deleting an new user group Update (+): Allows modifying an user group Suspend (=): Allows suspending an user group (All the users belonging to the group can not perform the login) Resume (=): Allows resuming an user group Assign (+): Assigns a user group to a role Unassign (+): Unassigns a user group from a role
namespace	http://edds.egos.esa/model
children	Create Delete Update Suspend Resume Assign Unassign
used by	element AccountRequest/Group
annotation	documentation Defines the operations which can be performed to manipulate a user group

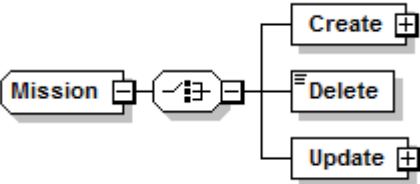
K.10 complexType GroupData

diagram	
namespace	http://edds.egos.esa/model
children	Name Description EnablingStatus UserList RoleList PwdChangeForced
used by	elements Group/Create MissionDetailsList/GroupDataList Group/Update

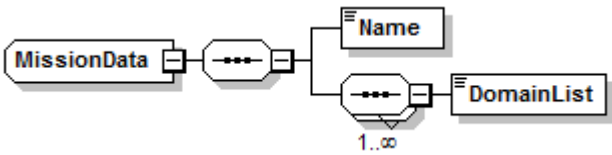
K.11 complexType GroupRoleAssignment

diagram	
namespace	http://edds.egos.esa/model
children	GroupName RoleList
used by	elements Group/Assign Group/Unassign

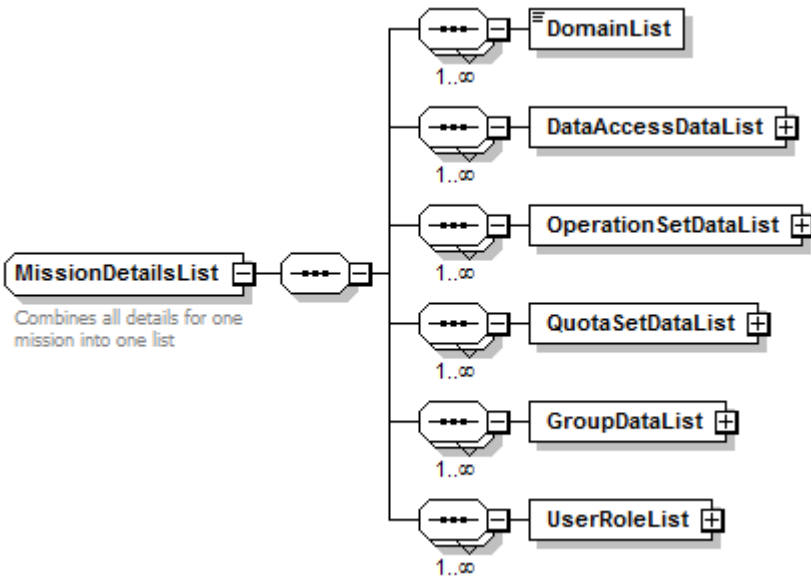
K.12 complexType Mission

diagram	
namespace	http://edds.egos.esa/model
children	Create Delete Update
used by	element AccountRequest/Mission

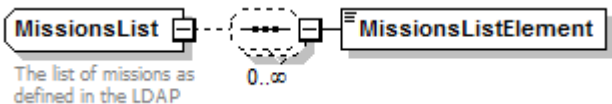
K.13 complexType MissionData

diagram	 The diagram shows the MissionData complex type. It consists of a root box labeled 'MissionData' connected to a sequence container (rectangle with three dots). This container has two children: a 'Name' element box and a 'DomainList' element box. The 'DomainList' box has a multiplicity of '1..∞' below it.
namespace	http://edds.egos.esa/model
children	Name DomainList
used by	elements Mission/Create Mission/Update

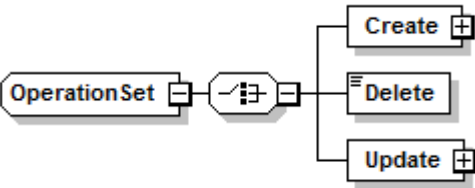
K.14 complexType MissionDetailsList

diagram	 The diagram shows the MissionDetailsList complex type. It consists of a root box labeled 'MissionDetailsList' with the annotation 'Combines all details for one mission into one list' below it. This box is connected to a sequence container (rectangle with three dots). This container has six children, each with a multiplicity of '1..∞' below it: 'DomainList', 'DataAccessDataList', 'OperationSetDataList', 'QuotaSetDataList', 'GroupDataList', and 'UserRoleList'. Each child box has a plus sign icon in its top right corner.
namespace	http://edds.egos.esa/model
children	DomainList DataAccessDataList OperationSetDataList QuotaSetDataList GroupDataList UserRoleList
used by	element MissionDetails
annotation	documentation Combines all details for one mission into one list

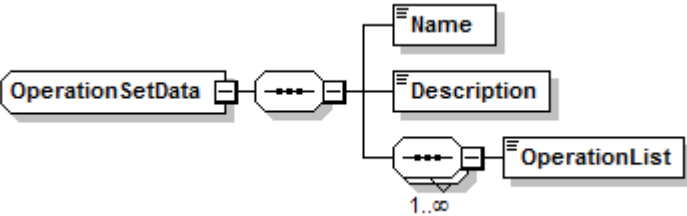
K.15 complexType MissionsList

diagram	 The diagram shows the MissionsList complex type. It consists of a root box labeled 'MissionsList' with the annotation 'The list of missions as defined in the LDAP' below it. This box is connected to a sequence container (rectangle with three dots) which has a multiplicity of '0..∞' below it. This container is connected to a 'MissionsListElement' element box.
namespace	http://edds.egos.esa/model
children	MissionsListElement
used by	element MissionsList
annotation	documentation The list of missions as defined in the LDAP

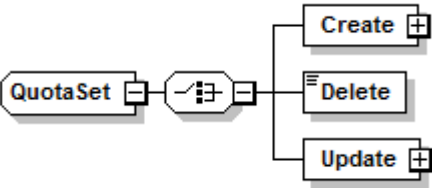
K.16 complexType OperationSet

diagram	
namespace	http://edds.egos.esa/model
children	Create Delete Update
used by	element AccountRequest/OperationSet

K.17 complexType OperationSetData

diagram	
namespace	http://edds.egos.esa/model
children	Name Description OperationList
used by	elements OperationSet/Create MissionDetailsList/OperationSetDataList OperationSet/Update

K.18 complexType QuotaSet

diagram	
namespace	http://edds.egos.esa/model
children	Create Delete Update
used by	element AccountRequest/QuotaSet

K.19 complexType QuotaSetData

diagram	The diagram illustrates the structure of the QuotaSetData complexType. It consists of a sequence of elements: Name, Description, a choice between MaxNumberOfRequestsPerPeriod and RequestQuotaPeriod, MaxNumberOfOngoingRequests, another choice between MaxAmountOfDataPerPeriod and DataQuotaPeriod, MaxDiskSpace, SpidsRestriction, ApidRestriction, ParameterNamesRestriction, TcCommandNamesRestriction, FARCFileTypeRestriction, FARCFileNameRestriction, FileSystemFolderRestriction, DataRangeRestriction, and DataDurationRestriction. Each element is accompanied by a descriptive text box explaining its function.
namespace	http://edds.egos.esa/model
children	Name Description MaxNumberOfRequestsPerPeriod RequestQuotaPeriod MaxNumberOfOngoingRequests MaxAmountOfDataPerPeriod DataQuotaPeriod MaxDiskSpace

	SpidsRestriction ApidsRestriction ParameterNamesRestriction TcCommandNamesRestriction FARCFileTypeRestriction FARCFileNameRestriction FileSystemFolderRestriction DataRangeRestriction DataDurationRestriction
used by	elements QuotaSet/Create UserQuotaDetails/QuotaLimits MissionDetailsList/QuotaSetDataList QuotaSet/Update

K.20 complexType QuotaSetRoleAssignment

diagram	
namespace	http://edds.egos.esa/model
children	QuotaSet RoleName
used by	elements QuotaSet/Assign QuotaSet/Unassign

K.21 complexType Role

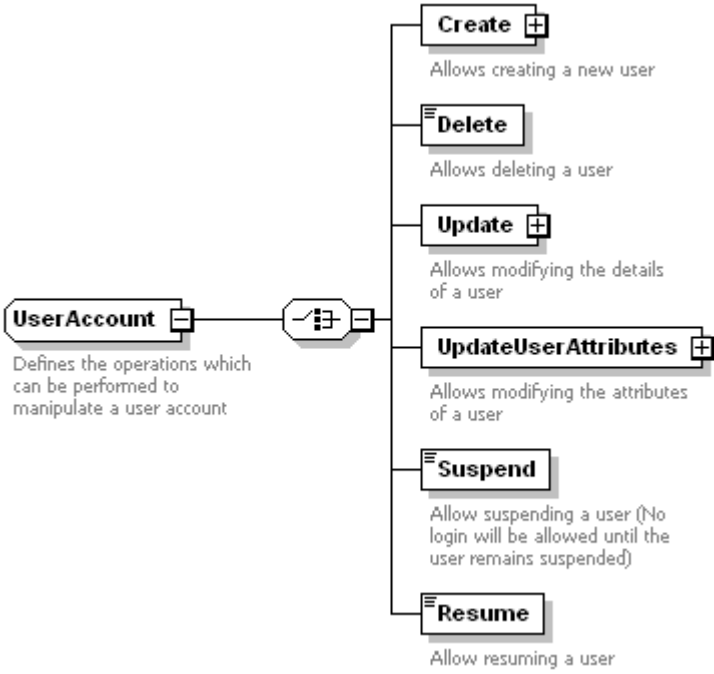
diagram	
namespace	http://edds.egos.esa/model
children	Create Delete Update Suspend Resume
used by	element AccountRequest/Role
annotation	documentation Defines the operations which can be performed to manipulate a user role

K.22 complexType RolesList

diagram	
namespace	http://edds.egos.esa/model
children	RoleElement
used by	element RolesList

annotation	documentation The list of roles a user has permission to use for the specified mission
------------	---

K.23 complexType UserAccount

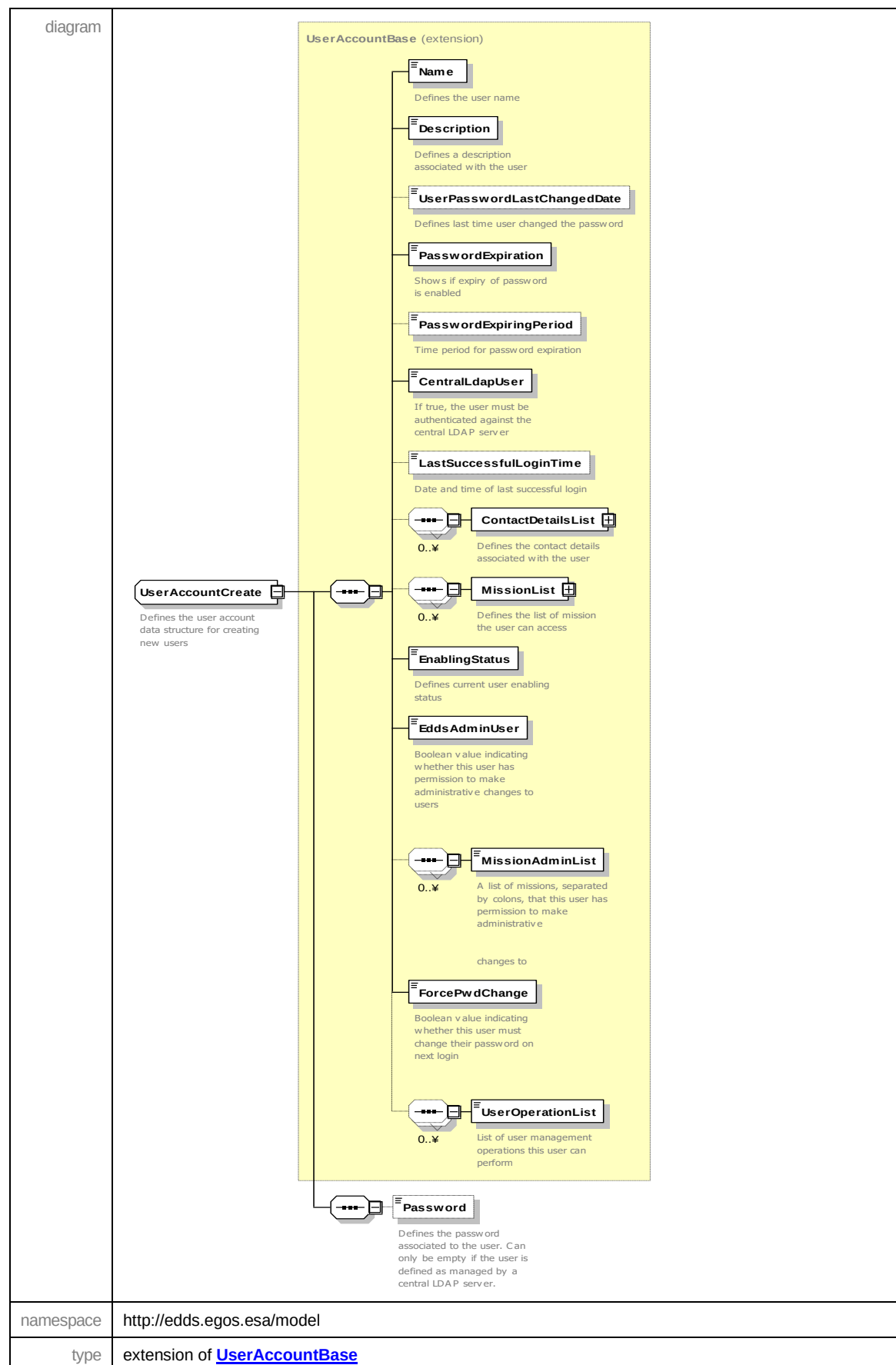
diagram	 The diagram shows a complexType UserAccount with a description "Defines the operations which can be performed to manipulate a user account". It is connected to a container element (octagon with a plus sign) which then branches into six child elements: Create (allows creating a new user), Delete (allows deleting a user), Update (allows modifying the details of a user), UpdateUserAttributes (allows modifying the attributes of a user), Suspend (allow suspending a user (No login will be allowed until the user remains suspended)), and Resume (allow resuming a user).
namespace	http://edds.egos.esa/model
children	Create Delete Update UpdateUserAttributes Suspend Resume
used by	element AccountRequest/UserAccount
annotation	documentation Defines the operations which can be performed to manipulate a user account

K.24 complexType UserAccountBase

diagram	UserAccountBase Defines the basic user account data structure without the password element Name Defines the user name Description Defines a description associated with the user UserPasswordLastChangedDate Defines last time user changed the password PasswordExpiration Shows if expiry of password is enabled PasswordExpiringPeriod Time period for password expiration CentralLdapUser If true, the user must be authenticated against the central LDAP server LastSuccessfulLoginTime Date and time of last successful login ContactDetailsList 0..* Defines the contact details associated with the user MissionList 0..* Defines the list of mission the user can access EnablingStatus Defines current user enabling status EddsAdminUser Boolean value indicating whether this user has permission to make administrative changes to users MissionAdminList 0..* A list of missions, separated by colons, that this user has permission to make administrative changes to ForcePwdChange Boolean value indicating whether this user must change their password on next login UserOperationList 0..* List of user management operations this user can perform
namespace	http://edds.egos.esa/model
children	Name Description UserPasswordLastChangedDate PasswordExpiration PasswordExpiringPeriod CentralLdapUser LastSuccessfulLoginTime ContactDetailsList MissionList EnablingStatus

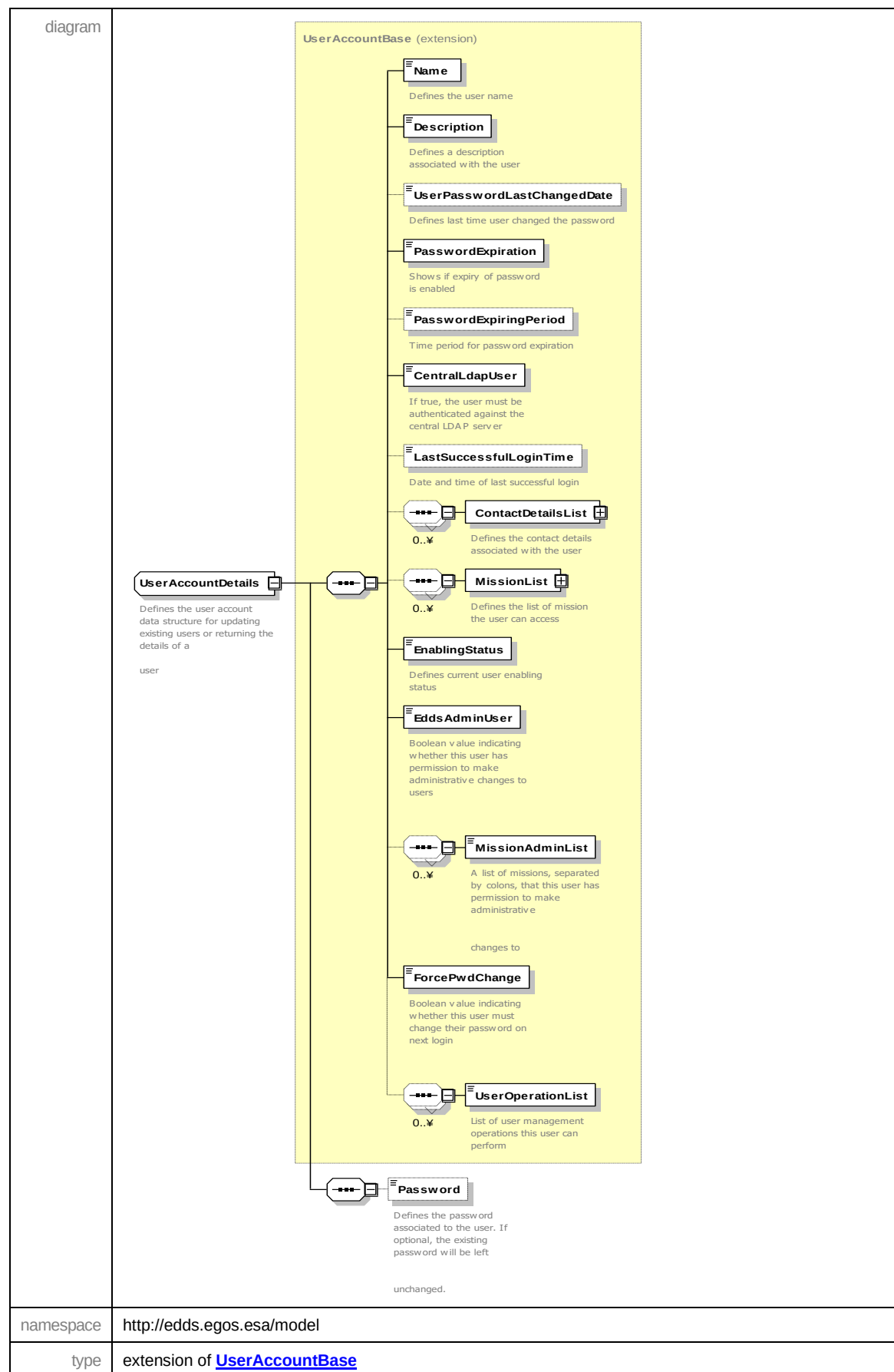
	EddsAdminUser MissionAdminList ForcePwdChange UserOperationList
used by	complexTypes UserAccountCreate UserAccountDetails
annotation	documentation Defines the basic user account data structure without the password element

K.25 complexType UserAccountCreate



properties	base UserAccountBase
children	Name Description UserPasswordLastChangedDate PasswordExpiration PasswordExpiringPeriod CentralLdapUser LastSuccessfulLoginTime ContactDetailsList MissionList EnablingStatus EddsAdminUser MissionAdminList ForcePwdChange UserOperationList Password
used by	element UserAccount/Create
annotation	documentation Defines the user account data structure for creating new users

K.26 complexType UserAccountDetails



properties	base UserAccountBase
children	Name Description UserPasswordLastChangedDate PasswordExpiration PasswordExpiringPeriod CentralLdapUser LastSuccessfulLoginTime ContactDetailsList MissionList EnablingStatus EddsAdminUser MissionAdminList ForcePwdChange UserOperationList Password
used by	elements UserAccount/Update UserAccountDetails
annotation	documentation Defines the user account data structure for updating existing users or returning the details of a user

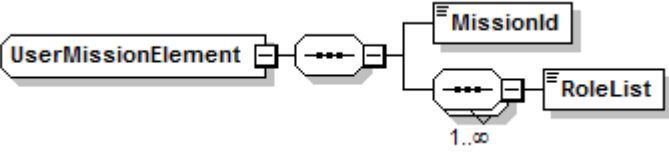
K.27 complexType UserAttributes

diagram	
namespace	http://edds.egos.esa/model
children	Name Description Password ForcePwdChange ContactDetailsList
used by	element UserAccount/UpdateUserAttributes
annotation	documentation Allows the currently logged in (non-EDDS Admin user) to update a subset of their account data

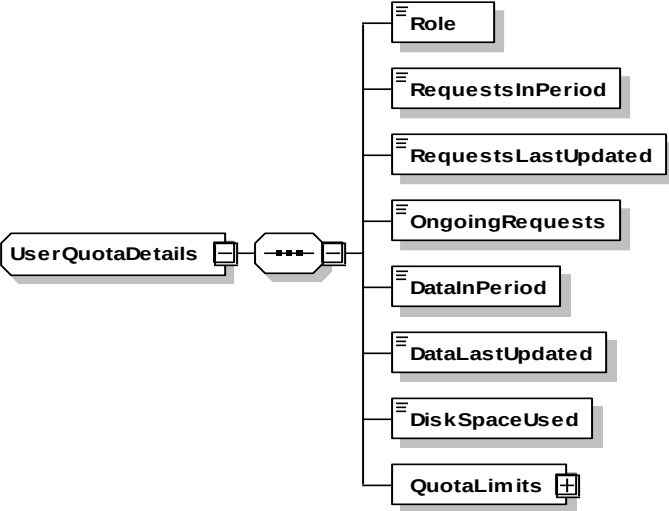
K.28 complexType UserMission

diagram	
namespace	http://edds.egos.esa/model
children	UserName MissionName
used by	element UserMission

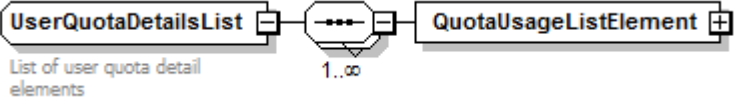
K.29 complexType UserMissionElement

diagram	
namespace	http://edds.egos.esa/model
children	MissionId RoleList
used by	element UserAccountBase/MissionList

K.30 complexType UserQuotaDetails

diagram	
namespace	http://edds.egos.esa/model
children	Role RequestsInPeriod RequestsLastUpdated OngoingRequests DataInPeriod DataLastUpdated DiskSpaceUsed QuotaLimits
used by	element UserQuotaDetailsList/QuotaUsageListElement

K.31 complexType UserQuotaDetailsList

diagram	
namespace	http://edds.egos.esa/model
children	QuotaUsageListElement
used by	element UserQuotaDetails
annotation	documentation List of user quota detail elements

K.32 complexType UserRole

diagram	
namespace	http://edds.egos.esa/model
children	Name Description OperationSet DataAccessSet QuotaSet Priority EnablingStatus
used by	elements Role/Create Role/Update MissionDetailsList/UserRoleList

K.33 complexType UsersList

diagram	
namespace	http://edds.egos.esa/model
children	UsersListElement
used by	element UsersList
annotation	documentation The list of users as defined in the LDAP

K.34 simpleType OperationElement

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	OperationSetData/OperationList	
facets	Kind	Value	annotation
	enumeration	CREATE_GROUP	
	enumeration	DELETE_GROUP	
	enumeration	UPDATE_GROUP	
	enumeration	ASSIGN_ROLE_TO_GROUP	
	enumeration	UNASSIGN_ROLE_FROM_GROUP	
	enumeration	SUSPEND_GROUP	

	enumeration	RESUME_GROUP
	enumeration	CREATE_ROLE
	enumeration	DELETE_ROLE
	enumeration	UPDATE_ROLE
	enumeration	SUSPEND_ROLE
	enumeration	RESUME_ROLE
	enumeration	UPDATE_ROLE_DEFAULT_CONTACT_DETAILS
	enumeration	CREATE_OPERATIONS_SET
	enumeration	DELETE_OPERATIONS_SET
	enumeration	UPDATE_OPERATIONS_SET
	enumeration	ASSIGN_OPERATIONS_SET_TO_ROLE
	enumeration	UNASSIGN_OPERATIONS_SET_FROM_ROLE
	enumeration	CREATE_DATA_ACCESS_SET
	enumeration	DELETE_DATA_ACCESS_SET
	enumeration	UPDATE_DATA_ACCESS_SET
	enumeration	ASSIGN_DATA_ACCESS_SET_TO_ROLE
	enumeration	UNASSIGN_DATA_ACCESS_SET_FROM_ROLE
	enumeration	CREATE_QUOTA_SET
	enumeration	DELETE_QUOTA_SET
	enumeration	UPDATE_QUOTA_SET
	enumeration	ASSIGN_QUOTA_SET_TO_ROLE
	enumeration	UNASSIGN_QUOTA_SET_FROM_ROLE
	enumeration	SCHEDULE_REQUEST
	enumeration	VIEW_LOG_FILE

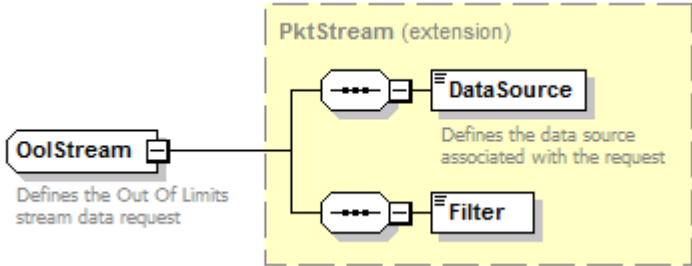
K.35 simpleType UserOperationElement

namespace	http://edds.egos.esa/model		
type	restriction of xs:string		
properties	base	xs:string	
used by	element	UserAccountBase/UserOperationList	
facets	Kind	Value	annotation
	enumeration	CREATE_USER_ACCOUNT	
	enumeration	DELETE_USER_ACCOUNT	
	enumeration	UPDATE_USER_ACCOUNT	
	enumeration	UPDATE_USER_PERSONAL_ATTRIBUTES	
	enumeration	SUSPEND_USER_ACCOUNT	
	enumeration	RESUME_USER_ACCOUNT	

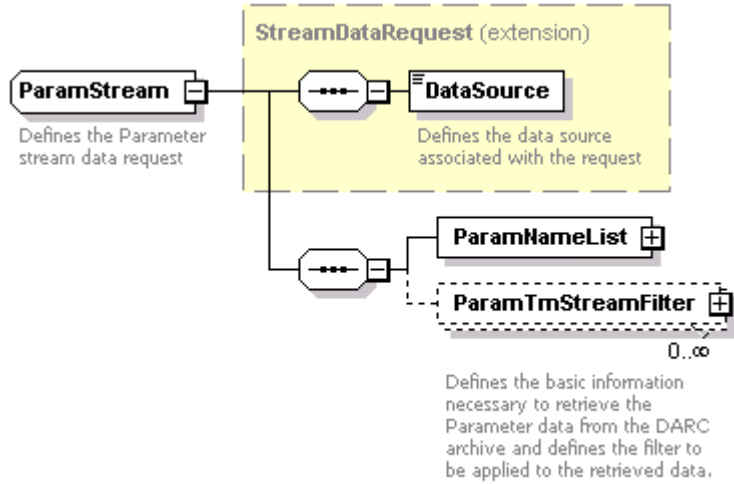
Appendix L Stream Request Data Type

This section of the document describes the complex types used for stream requests as defined in the streamrequest.xsd file.

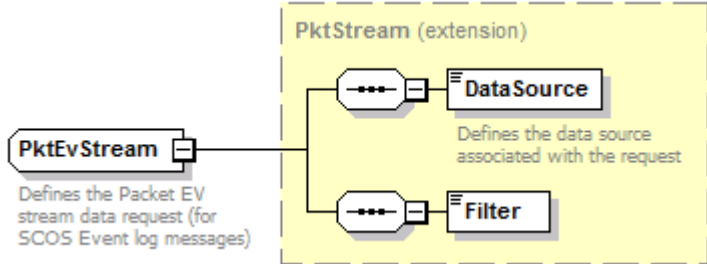
L.1 complexType OolStream

diagram	 The diagram shows the OolStream complex type, which defines the Out Of Limits stream data request. It is an extension of PktStream. The structure includes a DataSource element (defined as 'Defines the data source associated with the request') and a Filter element.
namespace	http://edds.egos.esa/model
type	extension of PktStream
properties	base PktStream
children	DataSource Filter
annotation	documentation Defines the Out Of Limits stream data request

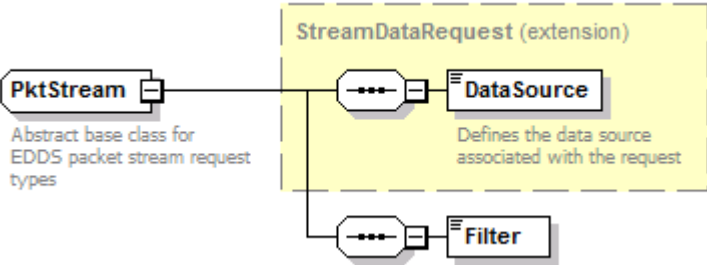
L.2 complexType ParamStream

diagram	 The diagram shows the ParamStream complex type, which defines the Parameter stream data request. It is an extension of StreamDataRequest. The structure includes a DataSource element (defined as 'Defines the data source associated with the request'), a ParamNameList element, and a ParamTmStreamFilter element (defined as 'Defines the basic information necessary to retrieve the Parameter data from the DARC archive and defines the filter to be applied to the retrieved data.'). The ParamTmStreamFilter element has a cardinality of 0..∞.
namespace	http://edds.egos.esa/model
type	extension of StreamDataRequest
properties	base StreamDataRequest
children	ParamNameList ParamTmStreamFilter
annotation	documentation Defines the Parameter stream data request

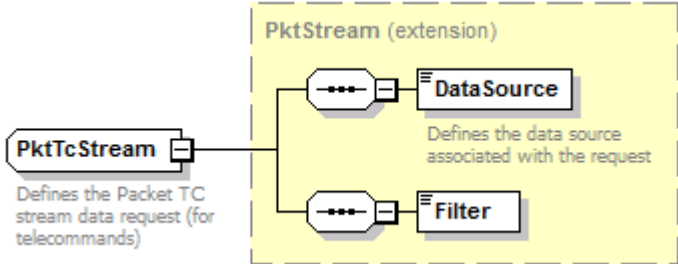
L.3 complexType PktEvStream

diagram	
namespace	http://edds.egos.esa/model
type	extension of PktStream
properties	base PktStream
children	DataSource Filter
annotation	documentation Defines the Packet EV stream data request (for SCOS Event log messages)

L.4 complexType PktStream

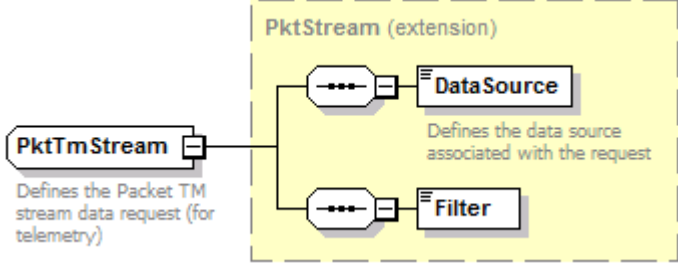
diagram	
namespace	http://edds.egos.esa/model
type	extension of StreamDataRequest
properties	base StreamDataRequest abstract true
used by	complexTypes OolStream PktEvStream PktTcStream PktTmStream
annotation	documentation Abstract base class for EDDS packet stream request types

L.5 complexType PktTcStream

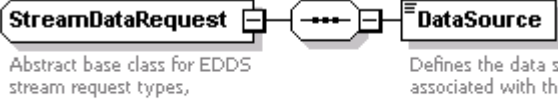
diagram	
namespace	http://edds.egos.esa/model
type	extension of PktStream
properties	base PktStream

children	DataSource Filter
annotation	documentation Defines the Packet TC stream data request (for telecommands)

L.6 complexType PktTmStream

diagram	
namespace	http://edds.egos.esa/model
type	extension of PktStream
properties	base PktStream
children	DataSource Filter
annotation	documentation Defines the Packet TM stream data request (for telemetry)

L.7 complexType StreamDataRequest

diagram	
namespace	http://edds.egos.esa/model
properties	abstract true
children	DataSource
used by	element StreamRequest/StreamDataRequest complexTypes ParamStream PktStream
annotation	documentation Abstract base class for EDDS stream request types, defines the specific stream data requested in the StreamRequest

L.8 complexType StreamRequest

diagram	
namespace	http://edds.egos.esa/model
children	Comment Schedule User StreamDataRequest
used by	element StreamRequestMessagePart/StreamRequest
annotation	documentation Defines the structure and content of a stream request.

L.9 complexType StreamRequestMessagePart

diagram	
namespace	http://edds.egos.esa/model
children	ContextPart StreamRequest
used by	element StreamRequestMessagePart
annotation	documentation Defines the stream request message the client submits to server

Appendix M EDDS Parameter Spreadsheet

M.1 TDRS like Spreadsheet

EDDS provides a sample XSLT file for the Telemetry Parameter batch request so that the TDRS spreadsheet format can be created from the retrieved XML. The file is called “Param.xsl” and can be modified to each mission’s needs. The file is deployed by default into the EDDS Server runtime directory within “config/XSLT”. The filename must match the report type. More details on the format of the generated TDRS spreadsheet file can be found in the file “TDRS External Interfaces Control Document” [RD-14]

The following fields are included in the output of the TDRS spreadsheet file, but are only filled with dummy values as the information cannot be obtained yet these fields are mandatory in the TDRS ICD:

- Retrieval ID (always set to 0);
- Execution ID (always set to 0);
- Expiration day (always set to 0).

The values can be changed by editing the XSLT file.

As the XSLT file must load the XML data into memory to be able to create the TDRS output file, EDDS also provides a native TDRS format that can work with much larger amounts of data. This format can be selected on the Parameter TM Format request and optionally the number of decimal places can be limited under the TDRS Options section. By default, EDDS will not limit the number of decimal places and so the data will be as received from the DARC/SMON. The restrictions above regarding the mandatory fields that cannot be populated remain in place, except the Retrieval ID is set to the EDDS Request ID.

M.2 Presto Plot Stylesheet

The style sheet for transforming XML results are mission specific and provided by the mission, and so considered out of the scope of this document.

Appendix N Google Protocol Buffers

N.1 Introduction

EDDS uses a software library called Google Protocol Buffers internally for use in passing messages between applications as well as for stream data. Google Protocol Buffers offers several advantages over the use of XML; in particular it is considerably faster and smaller than an XML message – an essential requirement for real-time streaming. More information on Google Protocol Buffers can be found here: <http://code.google.com/apis/protocolbuffers/>

EDDS uses the Google Protocol Buffers implementation found within the ActiveMQ library. However, the proto files are also compatible with the Protocol Buffers implementation from Google Inc. The proto files (which describe the data structure) are compiled into Java classes when compiling the software from source. The proto files can be found in the project edds-ws-common within src/main/protoFiles. The Maven plugin `org.apache.activemq.protobuf:activemq-protobuf` is used to compile the proto files.

EDDS uses the `toFramedByteArray()` and `fromFramedByteArray()` methods (of ActiveMQ protobuf implementation) in the compiled proto code for converting the data to a byte array for sending on the payload of a JMS message and reading the data back again on the receiver’s end. Several helper classes exist for reading the encoded byte arrays sent from the server to the client. The Java client API also uses these classes for making it easier for users of the API to access the information returned by the web server. The helper classes can be found in `esa.egos.edds.types` within edds-ws-common.

There is helper class `DarcParameterBinarySupport` available to simplify reading data from DARC parameter binary response. This class can be found from project edds-ws-client.

Here is a list of the proto files within EDDS and their purpose:

Proto file	Description
darcparameter.proto	A record containing information about a DARC parameter as placed on the DARC stream topic
farccatalogue.proto	Contains records containing information about files stored in the FARC
Catalogue.proto	Used for EDDS notification messaging
FsCatalogue.proto	Used for EDDS notification messaging about file change on filesystem
paramdef.proto	A record containing information about a DARC parameter definition
requestinfo.proto	A record containing information about a batch, stream or account request
systemlog.proto	A record containing information about a log message from EDDS
DarcParameterBinary.proto	A record containing information about DARC parameters in Parameter TM request response files in BINARY format
SMONParameter.proto	A record containing information about TM parameters in SMON Parameter request response files in BINARY format
EventRecordReport.proto	A record containing information about EV data in Event Record Report request response files in BINARY format
PacketRaw.proto	A record containing information about raw data in Packet TC Raw, Packet TM Raw and Packet EV Raw requests response files in BINARY format
PacketTCReport.proto	A record containing information about TC data in Packet TC Report request response files in BINARY format
PacketTMReport.proto	A record containing information about TM data in Packet TM Report request response files in BINARY format
PacketTMGapReport.proto	A record containing information about TM gap data in Packet TM Gap Report request response files in BINARY format
ParamValidity.proto	Definition of parameter validity as a separate proto file to be reused by multiple proto file definitions.
OOLReport.proto	A record containing information about out of limit data in Out of Limits Report request response files in BINARY format

Table 39 - List of the proto files within EDDS

Appendix O Code Examples

O.1 Parameter Definition Downloader

```
package com.github.user.parameter;

import java.io.BufferedOutputStream;
import java.io.Closeable;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.MalformedURLException;
import java.util.ArrayList;
import java.util.List;

import javax.xml.ws.WebServiceException;

import org.apache.activemq.protobuf.InvalidProtocolBufferException;

import esa.egos.edds.common.EddsException;
import esa.egos.edds.model.AckDelivery;
import esa.egos.edds.model.BatchRequest;
import esa.egos.edds.model.ContextPart;
import esa.egos.edds.model.DataCompression;
import esa.egos.edds.model.DataEncrypting;
import esa.egos.edds.model.DataRequest;
import esa.egos.edds.model.DataSource;
import esa.egos.edds.model.Delivery;
import esa.egos.edds.model.DownloadList;
import esa.egos.edds.model.EmailDelivery;
import esa.egos.edds.model.EmailList;
import esa.egos.edds.model.JobListIdPart;
import esa.egos.edds.model.ParamDefinition;
import esa.egos.edds.model.ParamDefinitionFormat;
import esa.egos.edds.model.ParamDefinitionFormatting;
import esa.egos.edds.model.PrivacyTag;
import esa.egos.edds.model.RequestInfoProto.RequestInfo.State;
import esa.egos.edds.model.RequestMessagePart;
import esa.egos.edds.model.RequestPostProcessing;
import esa.egos.edds.model.ResponseDataElement;
import esa.egos.edds.model.ScheduleOnce;
import esa.egos.edds.model.ServerDelivery;
import esa.egos.edds.model.User;
import esa.egos.edds.notifications.NotificationChannel;
import esa.egos.edds.service.AuthenticationFault;
import esa.egos.edds.service.AuthorizationFault;
import esa.egos.edds.service.PasswordExpiredFault;
import esa.egos.edds.service.RequestFault;
import esa.egos.edds.service.SessionFault;
import esa.egos.edds.types.RequestInfoRec;
import esa.egos.edds.ws.client.EddsService;
import esa.egos.edds.ws.client.notifications.NotificationListener;
import esa.egos.edds.ws.client.notifications.NotificationManager;
```

```
/**
 * Example class on how to request for parameter definitions as a batch
 * request, then download the response once it has been delivered
 */

public class ParameterDefinitionDownloader implements
NotificationListener < RequestInfoRec > {

    private final String username;
    private final String password;
    private final File downloadDirectory;
    private List < String > activeJobs = new ArrayList < String > ();
    private static final String SERVER_ENDPOINT_PROPERTY =
"edds.server.endpoint.url";
    private EddsService eddsService;

    /**
     * The constructor set mandatory fields as the EDDS Web Service
     * address, the login username and password, the preferred downloadDirectory
     */
    public ParameterDefinitionDownloader(String endPointAddress, String
username, String password, File downloadDirectory)
        throws MalformedURLException,
        WebServiceException,
        AuthenticationFault,
        PasswordExpiredFault,
        EddsException {
        if (!downloadDirectory.isDirectory()) {
            throw new EddsException("The directory: " +
downloadDirectory.getAbsolutePath() + " could not be found");
        }

        this.downloadDirectory = downloadDirectory;
        this.username = username;
        this.password = password;

        System.setProperty(SERVER_ENDPOINT_PROPERTY, endPointAddress);
        eddsService = new EddsService(endPointAddress);
        NotificationManager notificationManager = new
NotificationManager(eddsService, endPointAddress);
        eddsService.setNotificationManager(notificationManager);
        eddsService.connectToServer();
        checkLoggedIn();

        notificationManager.registerListener(NotificationChannel.STATUS_UPDATES,
this);
    }

    /**
     * The method executes login into EDDS Web Services
     * @throws MalformedURLException
     * @throws WebServiceException
     * @throws AuthenticationFault
     * @throws PasswordExpiredFault
     */
    private void checkLoggedIn() throws MalformedURLException,
WebServiceException,
```

```
AuthenticationFault,
PasswordExpiredFault {
    if (eddsService.hasSessionExpired()) {
        eddsService.logIn(username, password);
    }
}

/**
 * The method build a Parameter Definition request and send it to
EDDS Web Services.
 * The response is downloaded into the directory
 * @param mission
 * @param role
 * @throws MalformedURLException
 * @throws WebServiceException
 * @throws RequestFault
 * @throws SessionFault
 * @throws AuthorizationFault
 * @throws EddsException
 */
public void getParamDefinitions(String mission, String myDomain,
String role) throws MalformedURLException,
WebServiceException,
RequestFault,
SessionFault,
AuthorizationFault,
EddsException {
    // Build up the XML batch request in Java objects
    RequestMessagePart requestMessagePart = new RequestMessagePart();
    BatchRequest batchRequest = new BatchRequest();

    // creating the context: the following is the most common way to
build the context
    ContextPart contextPart = new ContextPart();
    // Domain doesn't apply to parameter definition requests
    contextPart.setDomainId(myDomain);
    contextPart.setMissionId(mission);
    contextPart.setPrivacyTag(PrivacyTag.PRIVATE);
    requestMessagePart.setContextPart(contextPart);

    User user = new User();
    user.setRole(role);
    user.setUserName(username);
    batchRequest.setUser(user);
    ScheduleOnce schedule = new ScheduleOnce();
    schedule.setImmediate("");
    batchRequest.setSchedule(schedule);
    DataRequest dataRequest = new DataRequest();

    ParamDefinition paramDefinition = new ParamDefinition();
    Delivery delivery = new Delivery();
    ServerDelivery serverDelivery = new ServerDelivery();
    delivery.setServerDelivery(serverDelivery);
    paramDefinition.setDelivery(delivery);
    ParamDefinitionFormatting paramDefFormat = new
ParamDefinitionFormatting();
    // setting of the preferred response format
```

```

        paramDefFormat.setFormat (ParamDefinitionFormat.XML);
        paramDefinition.setParamDefinitionFormat (paramDefFormat);
        RequestPostProcessing postProcessing = new
RequestPostProcessing();
        postProcessing.setXfdu (false);

        //set of the preferred response compression/encryption
        postProcessing.setDataCompression (DataCompression.NONE);
        postProcessing.setDataEncrypting (DataEncrypting.NONE);
        paramDefinition.setRequestPostProcessing (postProcessing);
        paramDefinition.setDataSource (DataSource.DARC);

        // setting the data request as parameter definition request
        dataRequest.setParamDefinition (paramDefinition);
        //setting the batch request as the customised data request
        batchRequest.setDataRequest (dataRequest);
        batchRequest.setComment ("Param definitions request");

        // setting the email for notification
        AckDelivery ackDelivery = new AckDelivery();
        EmailDelivery emailDelivery = new EmailDelivery();
        emailDelivery.setDefault (false);
        EmailList emailList = new EmailList();
        // Add any e-mail addresses required for delivery notifications
        emailList.getEmailListElement().add ("joe.bloggs@company.com");
        emailDelivery.setEmailList (emailList);
        ackDelivery.setEmailDelivery (emailDelivery);
        batchRequest.setAckDelivery (ackDelivery);
        requestMessagePart.setBatchRequest (batchRequest);

        // Send the batch request
        JobListIdPart jobListIdPart =
eddsService.batchRequest (requestMessagePart);

        // Only one job ID is given in this case as it's not scheduled on
a repeating basis
        if (jobListIdPart.getJobListIdPartElement().size() != 1) {
            throw new EddsException ("Unexpected number of job IDs
received! Received: " + jobListIdPart.getJobListIdPartElement().size());
        }
        synchronized (activeJobs) {

activeJobs.add (jobListIdPart.getJobListIdPartElement().get (0));
        }

        public void dispose() throws MalformedURLException,
WebServiceException {
            eddsService.logout();
            synchronized (activeJobs) {
                activeJobs.clear();
            }
        }

        @Override
        public void onNotification (RequestInfoRec notification, String
requestId) {

```



```

        System.out.println("onNotification method called");
        try {
            synchronized(activeJobs) {

                if (activeJobs.contains(notification.getRequestId()) &&
notification.getRequestState() == State.DELIVERED) {
                    System.out.println("Downloading response files for
request: " + notification.getRequestId());
                    DownloadList downloadList = new DownloadList();
                    downloadList.setJob(notification.getRequestId());
                    // Notification that the job is complete - now we can
download the job
                    List < ResponseDataElement > dataHandlers =
eddsService.getResponse(downloadList).getResponseDataElement();
                    // Save the response files
                    for (ResponseDataElement responseElem: dataHandlers)
{
                        String fileName =
downloadDirectory.getAbsolutePath() + File.separator +
responseElem.getFileName();
                        try (BufferedOutputStream bos = new
BufferedOutputStream(new FileOutputStream(new File(fileName)))) {
                            responseElem.getResponse().writeTo(bos);
                            System.out.println("Wrote file: " +
fileName);
                        } finally {
                            if (responseElem.getResponse() instanceof
Closeable) {
                                try {
                                    // This is necessary to avoid
creating many temporary files
                                    ((Closeable)
responseElem.getResponse()).close();
                                } catch (IOException e) {
                                    e.printStackTrace();
                                }
                            }
                        }

                        // Finally remove the job from the active list
                        activeJobs.remove(notification.getRequestId());
                    }
                }
            } catch (InvalidProtocolBufferException e) {
                e.printStackTrace();
            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (WebServiceException e) {
                e.printStackTrace();
            } catch (RequestFault e) {
                e.printStackTrace();
            } catch (SessionFault e) {
                e.printStackTrace();
            } catch (AuthorizationFault e) {
                e.printStackTrace();
            } catch (FileNotFoundException e) {

```

```

        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static void main(String args[]) {
    ParameterDefinitionDownloader paramsDefDownloader = null;
    String eddsEndPoint =
"http://localhost:8080/edds/EddsService?wsdl";
    String myUsername = "<myusername>";
    String myPassword = "<mypassword>";
    String myMission = "RTE53";
    String myDomain = "0";
    String myRole = "Roll";
    File myDownloadDirectory = new
File(System.getProperty("user.dir"), (myUsername + "_download"));

    try {
        paramsDefDownloader = new
ParameterDefinitionDownloader(eddsEndPoint, myUsername, myPassword,
myDownloadDirectory);
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (WebServiceException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (AuthenticationFault e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (PasswordExpiredFault e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (EddsException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

    System.out.println("Gracefully Instantiated the
ParamDefinitionDownloader Object.");

    try {
        paramsDefDownloader.getParamDefinitions(myMission, myDomain,
myRole);
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (WebServiceException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (RequestFault e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (SessionFault e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

```

        } catch (AuthorizationFault e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (EddsException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        System.out.println("Gracefully called the getParamDefinitions
method.");
    }
}

```

O.2 Darc Parameter Downloader

```

package com.github.user.parameter;

import java.io.BufferedOutputStream;
import java.io.Closeable;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.MalformedURLException;
import java.time.Instant;
import java.util.ArrayList;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;
import java.util.TimeZone;

import javax.xml.datatype.DatatypeConfigurationException;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.ws.WebServiceException;

import org.apache.activemq.protobuf.InvalidProtocolBufferException;

import esa.egos.edds.common.EddsException;
import esa.egos.edds.model.AckDelivery;
import esa.egos.edds.model.BatchRequest;
import esa.egos.edds.model.ContextPart;
import esa.egos.edds.model.DataCompression;
import esa.egos.edds.model.DataEncrypting;
import esa.egos.edds.model.DataRequest;
import esa.egos.edds.model.DataSource;
import esa.egos.edds.model.Delivery;
import esa.egos.edds.model.DeliveryRange;
import esa.egos.edds.model.DownloadList;
import esa.egos.edds.model.EmailDelivery;
import esa.egos.edds.model.EmailList;
import esa.egos.edds.model.JobListIdPart;
import esa.egos.edds.model.Param;
import esa.egos.edds.model.ParamNameList;
import esa.egos.edds.model.ParamTmFilter;
import esa.egos.edds.model.ParamTmFormat;

```

```

import esa.egos.edds.model.ParamTmFormatting;
import esa.egos.edds.model.PrivacyTag;
import esa.egos.edds.model.RequestMessagePart;
import esa.egos.edds.model.RequestPostProcessing;
import esa.egos.edds.model.ResponseDataElement;
import esa.egos.edds.model.ScheduleOnce;
import esa.egos.edds.model.ServerDelivery;
import esa.egos.edds.model.TimeFiltering;
import esa.egos.edds.model.TimeRange;
import esa.egos.edds.model.RequestInfoProto.RequestInfo.State;
import esa.egos.edds.notifications.NotificationChannel;
import esa.egos.edds.service.AuthenticationFault;
import esa.egos.edds.service.AuthorizationFault;
import esa.egos.edds.service.PasswordExpiredFault;
import esa.egos.edds.service.RequestFault;
import esa.egos.edds.service.SessionFault;
import esa.egos.edds.types.RequestInfoRec;
import esa.egos.edds.ws.client.EddsService;
import esa.egos.edds.ws.client.notifications.NotificationListener;
import esa.egos.edds.ws.client.notifications.NotificationManager;

public class DARCPParameterDownloader implements NotificationListener
<RequestInfoRec>
{

    private final String      username;
    private final String      password;
    private final File        downloadDirectory;
    private String emailAddress;
    private String mission;
    private String domain;
    private String role;
    private PrivacyTag privacyTag;
    private List <String>      activeJobs          = new
ArrayList <String>();
    private static final String SERVER_ENDPOINT_PROPERTY =
"edds.server.endpoint.url";
    private EddsService eddsService;
    private static DatatypeFactory df;

    static
    {
        try
        {
            // Create this once and re-use it for better
performance
            df = DatatypeFactory.newInstance();
        }
        catch (DatatypeConfigurationException dce)
        {
            throw new IllegalStateException("Exception while
obtaining DatatypeFactory instance", dce);
        }
    }

    public ContextPart createContextPart()
    {

```

```
    // Set the mission details. This is as configured in User
Management in the EDDS MMI
    ContextPart context = new ContextPart();
    context.setMissionId(mission);
    context.setDomainId(domain);
    context.setPrivacyTag(privacyTag);
    return context;
}

public TimeRange createTimeRange(Instant start, Instant end)
{
    TimeRange range = new TimeRange();
    GregorianCalendar cal = new GregorianCalendar();
    cal.setTimeZone(TimeZone.getTimeZone("GMT"));
    cal.setTimeInMillis(start.getEpochSecond()*1000 );

    range.setStartTime(df.newXMLGregorianCalendar(cal));

    cal = new GregorianCalendar();
    cal.setTimeZone(TimeZone.getTimeZone("GMT"));
    cal.setTimeInMillis(end.getEpochSecond()*1000 );

    range.setEndTime(df.newXMLGregorianCalendar(cal));

    return range;
}

public AckDelivery createAckDelivery()
{
    AckDelivery ack = new AckDelivery();

    EmailDelivery emailDelivery = new EmailDelivery();
    EmailList emailList = new EmailList();
    // Can also be left empty.
    emailList.getEmailListElement().add(emailAddress);
    emailDelivery.setEmailList(emailList);
    emailDelivery.setDefault(true);
    ack.setEmailDelivery(emailDelivery);
    return ack;
}

public Delivery createDelivery(String filename)
{
    Delivery delivery = new Delivery();
    ServerDelivery serverDelivery = new ServerDelivery();
    serverDelivery.setFileName(filename);
    delivery.setServerDelivery(serverDelivery);
    return delivery;
}

public RequestPostProcessing createPostProcessing(DataCompression
compressionChoice)
{
    RequestPostProcessing postProcess = new
RequestPostProcessing();
    postProcess.setDataCompression(compressionChoice);
}
```

```

        postProcess.setDataEncrypting(DataEncrypting.NONE);
        return postProcess;
    }

    public DARCParameterDownloader(String endPointAddress, String
username, String password, File downloadDirectory)
        throws MalformedURLException, WebServiceException,
AuthenticationFault, PasswordExpiredFault, EddsException {
        super();
        this.username = username;
        this.password = password;
        this.downloadDirectory = downloadDirectory;

        System.setProperty(SERVER_ENDPOINT_PROPERTY,
endPointAddress);
        eddsService = new EddsService(endPointAddress);
        NotificationManager notificationManager = new
NotificationManager(eddsService, endPointAddress);
        eddsService.setNotificationManager(notificationManager);
        eddsService.connectToServer();
        checkLoggedIn();

        notificationManager.registerListener(NotificationChannel.STATUS_U
PDATES, this);
    }

    private void checkLoggedIn() throws MalformedURLException,
WebServiceException, AuthenticationFault, PasswordExpiredFault
    {
        if (eddsService.hasSessionExpired())
        {
            eddsService.logIn(username, password);
        }
    }

    public void dispose() throws MalformedURLException,
WebServiceException
    {
        eddsService.logOut();
        synchronized (activeJobs)
        {
            activeJobs.clear();
        }
    }

    /**
     * Example for creating DARC Parameter request
     *
     * @param filename the preferred filename to use
     * @param start the start timestamp (millis)
     * @param end the end timestamp (millis)
     * @return the request to send to EDDS
     * @throws DatatypeConfigurationException
     */
    public RequestMessagePart createDARCParameterRequest(String
filename, Instant start, Instant end, ParamTmFormat dataFormat,
DataCompression compressionChoice, List <String> darcParameters)

```

```

{
    RequestMessagePart part = new RequestMessagePart();

    part.setContextPart(createContextPart());

    BatchRequest request = new BatchRequest();
    request.setComment("DARC param Example request");

    // Execute as soon as possible
    ScheduleOnce schedule = new ScheduleOnce();
    schedule.setImmediate("");
    request.setSchedule(schedule);

    // User credentials as defined in User Management
    esa.egos.edds.model.User user = new
esa.egos.edds.model.User();
    user.setUserName(username);
    user.setRole(role);
    request.setUser(user);

    DataRequest dataRequest = new DataRequest();

    Param param = new Param();

    ParamTmFilter paramTmFilter = new ParamTmFilter();

    paramTmFilter.setTimeRange(createTimeRange(start, end));

    paramTmFilter.setTimeFiltering(TimeFiltering.GENERATION_TIME);

    // Add parameter names to fetch, or * for everything
    (potentially not a good idea!)
    ParamNameList paramNameList = new ParamNameList();
    for (String userParam : darcParameters) {

        paramNameList.getParamNameListElement().add(userParam);
    }
    paramTmFilter.setParamNameList(paramNameList);

    // Optionally specify DARC dataspace. If not specified,
    the EDDS Server default will be used
    paramTmFilter.setDarcDataSpace("DEFAULT");

    param.setParamTmFilter(paramTmFilter);

    DeliveryRange deliveryRange = new DeliveryRange();
    deliveryRange.setNone("");
    paramTmFilter.setDeliveryRange(deliveryRange);

    ParamTmFormatting reportFormatting = new
ParamTmFormatting();
    // Use binary format for best performance
    reportFormatting.setFormat(dataFormat);

    param.setParamTmFormat(reportFormatting);

    param.setDelivery(createDelivery(filename));

```

```

        param.setRequestPostProcessing(createPostProcessing(compressionChoice));

        param.setDataSource(dataSource.DARC);

        dataRequest.setParam(param);
        request.setDataRequest(dataRequest);

        request.setAckDelivery(createAckDelivery());

        part.setBatchRequest(request);

        return part;
    }

    public void sendRequest (RequestMessagePart userRequest) throws
MalformedURLException, WebServiceException, RequestFault,
        SessionFault, AuthorizationFault, EddsException {
        // Send the batch request
        JobListIdPart jobListIdPart =
eddsService.batchRequest(userRequest);

        // Only one job ID is given in this case as it's not
scheduled on a repeating basis
        if (jobListIdPart.getJobListIdPartElement().size() != 1)
        {
            throw new EddsException("Unexpected number of job
IDs received! Received: " +
jobListIdPart.getJobListIdPartElement().size());
        }
        synchronized (activeJobs)
        {

            activeJobs.add(jobListIdPart.getJobListIdPartElement().get(0));
        }
    }

    @Override
    public void onNotification(RequestInfoRec notification, String
requestId)
    {
        System.out.println("onNotification method called");
        try
        {
            synchronized (activeJobs)
            {

                if
(activeJobs.contains(notification.getRequestId()) &&
notification.getRequestState() == State.DELIVERED)
                {
                    System.out.println("Downloading
response files for request: " + notification.getRequestId());
                    DownloadList downloadList = new
DownloadList();

```



```
        downloadList.setJob(notification.getRequestId());
        // Notification that the job is
complete - now we can download the job
        List <ResponseDataElement>
dataHandlers =
eddsService.getResponse(downloadList).getResponseDataElement();
        // Save the response files
        for (ResponseDataElement
responseElem : dataHandlers)
        {
            String fileName =
downloadDirectory.getAbsolutePath() + File.separator +
responseElem.getFileName();
            try (BufferedOutputStream
bos = new BufferedOutputStream(new FileOutputStream(new File(fileName))))
            {
                responseElem.getResponse().writeTo(bos);

                System.out.println("Wrote file: " + fileName);
            }
            finally
            {
                if
(responseElem.getResponse() instanceof Closeable)
                {
                    try
                    {
                        //
This is necessary to avoid creating many temporary files

                        ((Closeable) responseElem.getResponse()).close();
                    }
                    catch
                    {
                        (IOException e)

                        e.printStackTrace();
                    }
                }
            }
        }

        // Finally remove the job from the
active list

        activeJobs.remove(notification.getRequestId());
    }
}
catch (InvalidProtocolBufferException e)
{
    e.printStackTrace();
}
catch (MalformedURLException e)
{

```

```
        e.printStackTrace();
    }
    catch (WebServiceException e)
    {
        e.printStackTrace();
    }
    catch (RequestFault e)
    {
        e.printStackTrace();
    }
    catch (SessionFault e)
    {
        e.printStackTrace();
    }
    catch (AuthorizationFault e)
    {
        e.printStackTrace();
    }
    catch (FileNotFoundException e)
    {
        e.printStackTrace();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}

public String getEmailAddress() {
    return emailAddress;
}

public void setEmailAddress(String emailAddress) {
    this.emailAddress = emailAddress;
}

public String getMission() {
    return mission;
}

public void setMission(String mission) {
    this.mission = mission;
}

public String getDomain() {
    return domain;
}

public void setDomain(String domain) {
    this.domain = domain;
}

public String getRole() {
    return role;
}
```

```
public void setRole(String aRole) {
    this.role = aRole;
}

public PrivacyTag getPrivacyTag() {
    return privacyTag;
}

public void setPrivacyTag(PrivacyTag privacyTag) {
    this.privacyTag = privacyTag;
}

public static void main (String args[]) {
    DARCPParameterDownloader DARCPparamsDownloader = null;
    String eddsEndPoint =
"http://localhost:8080/edds/EddsService?wsdl";
    String myUsername = "<myusername>";
    String myPassword = "<mypassword>";
    String myMission = "RTE53";
    String myDomain = "0";
    String myEmailAddress = "<myemail>";
    String myFilename = "DARCPParamResult";
    PrivacyTag myPrivacyTag = PrivacyTag.PRIVATE;
    String myRole = "Roll";
    File myDownloadDirectory = new
File(System.getProperty("user.dir"), (myUsername+ "_download"));

    try {
        DARCPparamsDownloader = new
DARCPParameterDownloader(eddsEndPoint, myUsername, myPassword,
myDownloadDirectory);
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (WebServiceException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (AuthenticationFault e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (PasswordExpiredFault e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (EddsException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    System.out.println("Gracefully Instantiated the
DARCPParameterDownloader Object.");

    DARCPparamsDownloader.setMission(myMission);
    DARCPparamsDownloader.setDomain(myDomain);
    DARCPparamsDownloader.setPrivacyTag(myPrivacyTag);
    DARCPparamsDownloader.setRole(myRole);
    DARCPparamsDownloader.setEmailAddress(myEmailAddress);
}
```

```
        Instant myStart = Instant.parse("2014-11-20T07:00:00.00Z");
        Instant myEnd = Instant.parse("2014-12-30T17:00:00.00Z");

        ArrayList<String> myDarcParams = new ArrayList<String>();
        myDarcParams.add("NDW0000*");

        ParamTmFormat myPreferredFormat= ParamTmFormat.XML;
        DataCompression myPreferredCompression =
DataCompression.TAR; //Or: NONE, TARGZ

        RequestMessagePart darcMessage = null;

        try {
            darcMessage =
DARCPparamsDownloader.createDARCParameterRequest(myFilename, myStart,
myEnd, myPreferredFormat, myPreferredCompression, myDarcParams);
            System.out.println("Gracefully Instantiated the
RequestMessagePart Object.");
            DARCPparamsDownloader.sendRequest(darcMessage);
            System.out.println("Gracefully sent the
RequestMessagePart Object.");
        } catch (WebServiceException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (RequestFault e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SessionFault e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (AuthorizationFault e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (EddsException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        System.out.println("Gracefully called the
createDARCParameterRequest method.");
    }
}
```

Appendix P EDDS DevOps

EDDS DevOps use Gitlab for continuous integration, testing and deployment. The Gitlab pipeline is manually run from <https://gitlab.esa.int/edds/EDDS/>

It includes 3 stages – build, publish and deploy.

Build:

Builds the EDDS project artifacts.

Publish:

Publish the artifacts to the JFROG repository.

Deploy:

Deploys the EDDS artifacts to Kubernetes cluster and the EDDS components will be started automatically to running state.