Earth Observation
Mission CFI Software

EO_DATA_HANDLING
SOFTWARE USER MANUAL

Code:	EO-MA-DMS-GS-0007
Issue:	4.5
Date:	01/03/13

	Name	Function	Signature
Prepared by:	Juan Jose Borrego Bote	Project Engineers	
	Carlos Villanueva		
	Rubén Castro		
Checked by:	José Antonio González Abeytua	Project Manager	
Approved by:	José Antonio González Abeytua	Project Manager	

DEIMOS Space S.L.U.
Ronda de Poniente, 19
Edificio Fiteni VI, Portal 2, 2ª Planta
28760 Tres Cantos (Madrid), SPAIN
Tel.: +34 91 806 34 50
Fax: +34 91 806 34 51
E-mail: deimos@deimos-space.com

© DEIMOS Space S.L.U.

All Rights Reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of DEIMOS Space S.L.U. or ESA.

DOCUMENT INFORMATION

Contract Data		Classification	
Contract Number:	15583/01/NL/GS	Internal	
		Public	
Contract Issuer:	ESA / ESTEC	Industry	X
		Confidential	

External Distribution		
Name	Organisation	Copies

Electronic handling	
Word Processor:	LibreOffice 3.6.4
Archive Code:	P/SUM/DMS/01/026-044
Electronic file name:	eo-ma-dms-gs-007-10

DOCUMENT STATUS LOG

Issue	Change Description	Date	Approval
3.4	New document. Library first version. Issue number corresponds to CFI library issue	18/11/05	
3.5	• Maintenance release. • New features: - function xd_xml_validate	26/05/06	
3.6	• Maintenance release • New features: - Validator function and executable for XML files (xd_xml_validate and xml_validator)	24/11/06	
3.7	• Maintenance release • New features: - Function expcfi_check_libs - Library version for MAC OS X on Intel (32 and 64-bits)	13/07/07	
3.7.2	• Maintenance release • New features: - Reading and writing functions for TLE - New format for orbit files: reference frame added to the variable header.	31/07/08	
4.0	• Maintenance release • Reading function for the numerical propagator configuration file	16/01/09	
4.1	• Maintenance release • New section added: 8 List of schema's versions • New features: - GETASSEv2 DEM supported	07/05/10	
4.2	• Maintenance release • New features: - New format for the OSF to support curved MLST - New DEM configuration file	31/01/11	
4.3	• Maintenance release • New features: - Support for reading new IERS bulletins A and B - New functions to decimate orbit and attitude data: xd_orbit_file_decimate and xd_attitude_file_decimate	06/02/12	

4.4	• Maintenance release • New features: - New tags in DEM configuration for DEM cache	05/07/12	
4.5	• Maintenance release • New features: - New tags in DEM configuration for mini tiles and geoid compuation. - EarthCare filenames compliant with FFS 2.0.	01/03/13	

TABLE OF CONTENTS

DOCUMENT INFORMATION....2

DOCUMENT STATUS LOG....3

TABLE OF CONTENTS....5

LIST OF TABLES....17

1. SCOPE....22

2. ACRONYMS, NOMENCLATURE AND TERMINOLOGY....23

2.1. Acronyms....23

2.2. Nomenclature....23

2.3. Note on Terminology....24

3. APPLICABLE AND REFERENCE DOCUMENTS....25

3.1. Applicable Documents....25

3.2. Reference Documents....25

4. INTRODUCTION....26

4.1. Functions Overview....26

- 4.1.1. Reading routines....26
- 4.1.2. Writing routines....26
- 4.1.3. Functions to free memory....27
- 4.1.4. Validation of XML files....27

4.2. Reading and writing files....27

4.3. Memory usage....28

5. LIBRARY INSTALLATION....29

6. LIBRARY USAGE....30

6.1. Usage hints....32

6.2. General Enumerations............ 32

6.3. Data Structures............ 37

7. CFI FUNCTIONS DESCRIPTION............ 54

7.1. xd_read_fhr............ 55
7.1.1. Overview............ 55
7.1.2. Calling interface............ 55
7.1.3. Input parameters............ 55
7.1.4. Output parameters............ 55
7.1.5. Warnings and errors............ 56

7.2. xd_read_bulletin............ 57
7.2.1. Overview............ 57
7.2.2. Calling interface............ 57
7.2.3. Input parameters............ 57
7.2.4. Output parameters............ 57
7.2.5. Warnings and errors............ 58

7.3. xd_read_bulletin_2............ 59
7.3.1. Overview............ 59
7.3.2. Calling interface............ 59
7.3.3. Input parameters............ 59
7.3.4. Output parameters............ 59
7.3.5. Warnings and errors............ 60

7.4. xd_free_bulletin............ 61
7.4.1. Overview............ 61
7.4.2. Calling interface............ 61
7.4.3. Input parameters............ 61
7.4.4. Output parameters............ 61

7.5. xd_read_orbit_file............ 62
7.5.1. Overview............ 62
7.5.2. Calling interface............ 62
7.5.3. Input parameters............ 62
7.5.4. Output parameters............ 63
7.5.5. Warnings and errors............ 64

7.6. xd_free_orbit_file........ 65
7.6.1. Overview........ 65
7.6.2. Calling interface........ 65
7.6.3. Input parameters........ 65
7.6.4. Output parameters........ 65

7.7. xd_read_doris........ 66
7.7.1. Overview........ 66
7.7.2. Calling interface........ 66
7.7.3. Input parameters........ 66
7.7.4. Output parameters........ 67
7.7.5. Warnings and errors........ 67

7.8. xd_free_doris........ 69
7.8.1. Overview........ 69
7.8.2. Calling interface........ 69
7.8.3. Input parameters........ 69
7.8.4. Output parameters........ 69

7.9. xd_read_doris_header........ 70
7.9.1. Overview........ 70
7.9.2. Calling interface........ 70
7.9.3. Input parameters........ 70
7.9.4. Output parameters........ 70
7.9.5. Warnings and errors........ 71

7.10. xd_read_osf........ 72
7.10.1. Overview........ 72
7.10.2. Calling interface........ 72
7.10.3. Input parameters........ 72
7.10.4. Output parameters........ 72
7.10.5. Warnings and errors........ 73

7.11. xd_free_osf........ 75
7.11.1. Overview........ 75
7.11.2. Calling interface........ 75
7.11.3. Input parameters........ 75
7.11.4. Output parameters........ 75

7.12. xd_read_sdf... 76
7.12.1. Overview... 76
7.12.2. Calling interface... 76
7.12.3. Input parameters... 76
7.12.4. Output parameters... 76
7.12.5. Warnings and errors... 77

7.13. xd_free_sdf... 78
7.13.1. Overview... 78
7.13.2. Calling interface... 78
7.13.3. Input parameters... 78
7.13.4. Output parameters... 78

7.14. xd_read_stf... 79
7.14.1. Overview... 79
7.14.2. Calling interface... 79
7.14.3. Input parameters... 79
7.14.4. Output parameters... 79
7.14.5. Warnings and errors... 80

7.15. xd_free_stf... 82
7.15.1. Overview... 82
7.15.2. Calling interface... 82
7.15.3. Input parameters... 82
7.15.4. Output parameters... 82

7.16. xd_read_stf_vhr... 83
7.16.1. Overview... 83
7.16.2. Calling interface... 83
7.16.3. Input parameters... 83
7.16.4. Output parameters... 83
7.16.5. Warnings and errors... 84

7.17. xd_free_stf_vhr... 86
7.17.1. Overview... 86
7.17.2. Calling interface... 86
7.17.3. Input parameters... 86
7.17.4. Output parameters... 86

7.18. xd_read_att... 87
7.18.1. Overview... 87
7.18.2. Calling interface... 87
7.18.3. Input parameters... 87
7.18.4. Output parameters... 87
7.18.5. Warnings and errors... 88

7.19. xd_free_att... 90
7.19.1. Overview... 90
7.19.2. Calling interface... 90
7.19.3. Input parameters... 90
7.19.4. Output parameters... 90

7.20. xd_read_star_tracker... 91
7.20.1. Overview... 91
7.20.2. Calling interface... 91
7.20.3. Input parameters... 91
7.20.4. Output parameters... 92
7.20.5. Warnings and errors... 92

7.21. xd_free_star_tracker... 94
7.21.1. Overview... 94
7.21.2. Calling interface... 94
7.21.3. Input parameters... 94
7.21.4. Output parameters... 94

7.22. xd_read_star_tracker_conf_file... 95
7.22.1. Overview... 95
7.22.2. Calling interface... 95
7.22.3. Input parameters... 95
7.22.4. Output parameters... 95
7.22.5. Warnings and errors... 96

7.23. xd_read_dem... 97
7.23.1. Overview... 97
7.23.2. Calling interface... 97
7.23.3. Input parameters... 97
7.23.4. Output parameters... 98

7.23.5. Warnings and errors......98

7.24. xd_free_dem......100

7.24.1. Overview......100
7.24.2. Calling interface......100
7.24.3. Input parameters......100
7.24.4. Output parameters......100

7.25. xd_read_dem_config_file......101

7.25.1. Overview......101
7.25.2. Calling interface......101
7.25.3. Input parameters......101
7.25.4. Output parameters......101
7.25.5. Warnings and errors......102

7.26. xd_read_zone......103

7.26.1. Overview......103
7.26.2. Calling interface......103
7.26.3. Input parameters......103
7.26.4. Output parameters......104
7.26.5. Warnings and errors......104

7.27. xd_free_zone......106

7.27.1. Overview......106
7.27.2. Calling interface......106
7.27.3. Input parameters......106
7.27.4. Output parameters......106

7.28. xd_read_zone_file......107

7.28.1. Overview......107
7.28.2. Calling interface......107
7.28.3. Input parameters......107
7.28.4. Output parameters......107
7.28.5. Warnings and errors......108

7.29. xd_free_zone_file......110

7.29.1. Overview......110
7.29.2. Calling interface......110
7.29.3. Input parameters......110

7.29.4. Output parameters.. 110

7.30. xd_read_zone_id.. 111

7.30.1. Overview.. 111

7.30.2. Calling interface.. 111

7.30.3. Input parameters.. 111

7.30.4. Output parameters.. 111

7.30.5. Warnings and errors.. 112

7.31. xd_free_zone_id.. 114

7.31.1. Overview.. 114

7.31.2. Calling interface.. 114

7.31.3. Input parameters.. 114

7.31.4. Output parameters.. 114

7.32. xd_read_station.. 115

7.32.1. Overview.. 115

7.32.2. Calling interface.. 115

7.32.3. Input parameters.. 115

7.32.4. Output parameters.. 115

7.32.5. Warnings and errors.. 116

7.33. xd_read_station_file.. 118

7.33.1. Overview.. 118

7.33.2. Calling interface.. 118

7.33.3. Input parameters.. 118

7.33.4. Output parameters.. 118

7.33.5. Warnings and errors.. 119

7.34. xd_free_station_file.. 121

7.34.1. Overview.. 121

7.34.2. Calling interface.. 121

7.34.3. Input parameters.. 121

7.34.4. Output parameters.. 121

7.35. xd_read_station_id.. 122

7.35.1. Overview.. 122

7.35.2. Calling interface.. 122

7.35.3. Input parameters.. 122

7.35.4. Output parameters.......... 122
7.35.5. Warnings and errors.......... 123

7.36. xd_free_station_id.......... 125
7.36.1. Overview.......... 125
7.36.2. Calling interface.......... 125
7.36.3. Input parameters.......... 125
7.36.4. Output parameters.......... 125

7.37. xd_read_star.......... 126
7.37.1. Overview.......... 126
7.37.2. Calling interface.......... 126
7.37.3. Input parameters.......... 126
7.37.4. Output parameters.......... 126
7.37.5. Warnings and errors.......... 127

7.38. xd_read_star_file.......... 128
7.38.1. Overview.......... 128
7.38.2. Calling interface.......... 128
7.38.3. Input parameters.......... 128
7.38.4. Output parameters.......... 128
7.38.5. Warnings and errors.......... 129

7.39. xd_read_star_id.......... 130
7.39.1. Overview.......... 130
7.39.2. Calling interface.......... 130
7.39.3. Input parameters.......... 130
7.39.4. Output parameters.......... 130
7.39.5. Warnings and errors.......... 131

7.40. xd_read_tle.......... 132
7.40.1. Overview.......... 132
7.40.2. Calling interface.......... 132
7.40.3. Input parameters.......... 132
7.40.4. Output parameters.......... 132
7.40.5. Warnings and errors.......... 133

7.41. xd_free_tle.......... 135
7.41.1. Overview.......... 135

7.41.2. Calling interface........135
7.41.3. Input parameters........135
7.41.4. Output parameters........135

7.42. xd_read_precise_propag_file........136
7.42.1. Overview........136
7.42.2. Calling interface........136
7.42.3. Input parameters........136
7.42.4. Output parameters........136
7.42.5. Warnings and errors........137

7.43. xd_write_orbit_file........139
7.43.1. Overview........139
7.43.2. Calling interface........139
7.43.3. Input parameters........139
7.43.4. Output parameters........140
7.43.5. Warnings and errors........140

7.44. xd_write_osf........142
7.44.1. Overview........142
7.44.2. Calling interface........142
7.44.3. Input parameters........142
7.44.4. Output parameters........142
7.44.5. Warnings and errors........143

7.45. xd_write_doris........144
7.45.1. Overview........144
7.45.2. Calling interface........144
7.45.3. Input parameters........144
7.45.4. Output parameters........144
7.45.5. Warnings and errors........145

7.46. xd_write_stf........146
7.46.1. Overview........146
7.46.2. Calling interface........146
7.46.3. Input parameters........146
7.46.4. Output parameters........146
7.46.5. Warnings and errors........147

7.47. xd_write_att........ 148
7.47.1. Overview........ 148
7.47.2. Calling interface........ 148
7.47.3. Input parameters........ 148
7.47.4. Output parameters........ 149
7.47.5. Warnings and errors........ 149

7.48. xd_write_tle........ 151
7.48.1. Overview........ 151
7.48.2. Calling interface........ 151
7.48.3. Input parameters........ 151
7.48.4. Output parameters........ 151
7.48.5. Warnings and errors........ 152

7.49. xd_xml_validate........ 153
7.49.1. Overview........ 153
7.49.2. Calling interface........ 153
7.49.3. Input parameters........ 153
7.49.4. Output parameters........ 154
7.49.5. Warnings and errors........ 154
7.49.6. Executable program........ 155

7.50. xd_select_schema 157
7.50.1. Overview 157
7.50.2. Calling interface 157
7.50.3. Input parameters........ 157
7.50.4. Output parameters........ 157
7.50.5. Warnings and errors 158

7.51. xd_orbit_file_decimate........ 159
7.51.1. Overview........ 159
7.51.2. Calling interface........ 159
7.51.3. Input parameters........ 159
7.51.4. Output parameters........ 160
7.51.5. Warnings and errors........ 161

7.52. xd_attitude_file_decimate........ 162
7.52.1. Overview........ 162

7.52.2. Calling interface....162
7.52.3. Input parameters....162
7.52.4. Output parameters....163
7.52.5. Warnings and errors....164

8. LIST OF SCHEMA'S VERSIONS....165

9. FILES FORMAT SPECIFICATION....173

9.1. Fixed Header....173
9.1.1. Format....173
9.1.2. Example....175

9.2. Predicted Orbit files....176
9.2.1. Format....176
9.2.2. Example....178

9.3. Restituted Orbit files....179

9.4. Doris Preliminary/Precise files....179

9.5. Orbit Scenario files....179
9.5.1. Format....179
9.5.2. Example....182

9.6. Orbit Event files....184
9.6.1. Format....184
9.6.2. Example....184

9.7. DORIS Navigator files....186

9.8. Star Tracker files....186

9.9. Satellite Configuration File....187
9.9.1. Format....187
9.9.2. File Example....188

9.10. Attitude File....190
9.10.1. Format....190
9.10.2. File Example with angles....192
9.10.3. File Example with quaternions....193

9.11. Star tracker configuration File....196

9.11.1. Format....196
9.11.2. File Example....197

9.12. DEM Configuration File....200
9.12.1. Format....200
9.12.2. File Example....202

9.13. Swath Definition File....202
9.13.1. Format....202
9.13.2. File Example....209

9.14. Swath Template File....212
9.14.1. Format....212
9.14.2. File Example....214

9.15. Zone Database File....216
9.15.1. Format....216
9.15.2. File Example....217

9.16. Station Database File....220
9.16.1. Format....220
9.16.2. File Example....221

9.17. TLE File....224

9.18. Precise Propagator Configuration File....225
9.18.1. Format....225
9.18.2. File Example....226

10. RUNTIME PERFORMANCES....227

11. LIBRARY PRECAUTIONS....230

12. KNOWN PROBLEMS....231

LIST OF TABLES

Table 1: CFI functions included within EO_DATA_HANDLING library....31
Table 2: Enumerations within EO_DATA_HANDLING library....32
Table 3: EO_DATA_HANDLING Structures....37
Table 4: Input parameters of xd_read_fhr function....55
Table 5: Output parameters of xd_read_fhr function....55
Table 6: Error messages of xd_read_fhr function....56
Table 7: Input parameters of xd_read_bulletin function....57

Table 8: Output parameters of xd_read_bulletin function....57
Table 9: Error messages of xd_read_bulletin function....58
Table 10: Input parameters of xd_read_bulletin_2 function....59
Table 11: Output parameters of xd_read_bulletin_2 function....59
Table 12: Error messages of xd_read_bulletin_2 function....60
Table 13: Input parameters of xd_free_bulletin function....61
Table 14: Input parameters of xd_read_orbit_file function....62
Table 15: Output parameters of xd_read_orbit_file function....63
Table 16: Error messages of xd_read_orbit_file function....64
Table 17: Input parameters of xd_free_orbit_file function....65
Table 18: Input parameters of xd_read_doris function....66
Table 19: Output parameters of xd_read_doris function....67
Table 20: Error messages of xd_read_doris function....68
Table 21: Input parameters of xd_free_doris function....69
Table 22: Input parameters of xd_read_doris_header function....70
Table 23: Output parameters of xd_read_doris_header function....71
Table 24: Error messages of xd_read_doris function....71
Table 25: Input parameters of xd_read_osf function....72
Table 26: Output parameters of xd_read_osf function....72
Table 27: Error messages of xd_read_osf function....73
Table 28: Input parameters of xd_free_osf function....75
Table 29: Input parameters of xd_read_sdf function....76
Table 30: Output parameters of xd_read_sdf function....76
Table 31: Error messages of xd_read_sdf function....77
Table 32: Input parameters of xd_free_sdf function....78
Table 33: Input parameters of xd_read_stf function....79
Table 34: Output parameters of xd_read_stf function....80
Table 35: Error messages of xd_read_stf function....80
Table 36: Input parameters of xd_free_stf function....82
Table 37: Input parameters of xd_read_stf_vhr function....83
Table 38: Output parameters of xd_read_stf_vhr function....84
Table 39: Error messages of xd_read_stf_vhr function....84
Table 40: Input parameters of xd_free_stf_vhr function....86
Table 41: Input parameters of xd_read_att function....87
Table 42: Output parameters of xd_read_att function....88
Table 43: Error messages of xd_read_att function....88
Table 44: Input parameters of xd_free_att function....90
Table 45: Input parameters of xd_read_star_tracker function....91
Table 46: Output parameters of xd_read_star_tracker function....92
Table 47: Error messages of xd_read_star_tracker function....93
Table 48: Input parameters of xd_free_star_tracker function....94
Table 49: Input parameters of xd_read_star_tracker_conf_file function....95
Table 50: Output parameters of xd_read_star_tracker_conf_file function....96
Table 51: Error messages of xd_read_star_tracker_conf_file function....96
Table 52: Input parameters of xd_read_dem function....97
Table 53: Output parameters of xd_read_dem function....98
Table 54: Error messages of xd_read_dem function....98
Table 55: Input parameters of xd_free_dem function....100
Table 56: Input parameters of xd_read_dem_config_file function....101
Table 57: Output parameters of xd_read_dem_config_file function....102
Table 58: Error messages of xd_read_dem_config_file function....102

Table 59: Input parameters of xd_read_zone function....103
Table 60: Output parameters of xd_read_zone function....104
Table 61: Error messages of xd_read_zone function....104
Table 62: Input parameters of xd_free_zone function....106
Table 63: Input parameters of xd_read_zone_file function....107
Table 64: Output parameters of xd_read_zone_file function....108
Table 65: Error messages of xd_read_zone_file function....108
Table 66: Input parameters of xd_free_zone_file function....110
Table 67: Input parameters of xd_read_zone_id function....111
Table 68: Output parameters of xd_read_zone_id function....112
Table 69: Error messages of xd_read_zone_id function....112
Table 70: Input parameters of xd_free_zone_id function....114
Table 71: Input parameters of xd_read_station function....115
Table 72: Output parameters of xd_read_station function....116
Table 73: Error messages of xd_read_station function....116
Table 74: Input parameters of xd_read_station_file function....118
Table 75: Output parameters of xd_read_station_file function....119
Table 76: Error messages of xd_read_station_file function....119
Table 77: Input parameters of xd_free_station_file function....121
Table 78: Input parameters of xd_read_station_id function....122
Table 79: Output parameters of xd_read_station_id function....123
Table 80: Error messages of xd_read_station_id function....123
Table 81: Input parameters of xd_free_station_id function....125
Table 82: Input parameters of xd_read_star function....126
Table 83: Output parameters of xd_read_star function....127
Table 84: Error messages of xd_read_star function....127
Table 85: Input parameters of xd_read_star_file function....128
Table 86: Output parameters of xd_read_star_file function....129
Table 87: Error messages of xd_read_star_file function....129
Table 88: Input parameters of xd_read_star_id function....130
Table 89: Output parameters of xd_read_star_id function....131
Table 90: Error messages of xd_read_star_id function....131
Table 91: Input parameters of xd_read_tle function....132
Table 92: Output parameters of xd_read_tle function....133
Table 93: Error messages of xd_read_tle function....133
Table 94: Input parameters of xd_free_tle function....135
Table 95: Input parameters of xd_read_precise_propag function....136
Table 96: Output parameters of xd_read_precise_propag function....137
Table 97: Error messages of xd_read_precise_propag function....137
Table 98: Input parameters of xd_write_orbit_file function....139
Table 99: Output parameters of xd_write_orbit_file function....140
Table 100: Error messages of xd_write_orbit_file function....140
Table 101: Input parameters of xd_write_osf function....142
Table 102: Output parameters of xd_write_osf function....143
Table 103: Error messages of xd_write_osf function....143
Table 104: Input parameters of xd_write_doris function....144
Table 105: Output parameters of xd_write_doris function....145
Table 106: Error messages of xd_write_doris function....145
Table 107: Input parameters of xd_write_stf function....146
Table 108: Output parameters of xd_write_stf function....147
Table 109: Error messages of xd_write_stf function....147

Table 110: Input parameters of xd_write_att function....148
Table 111: Output parameters of xd_write_att function....149
Table 112: Error messages of xd_write_att function....149
Table 113: Input parameters of xd_write_tle function....151
Table 114: Output parameters of xd_write_tle function....151
Table 115: Error messages of xd_write_tle function....152
Table 116: Input parameters of xd_xml_validate function....153
Table 117: Output parameters of xd_xml_validate function....154
Table 118: Error messages of xd_xml_validate function....154
Table 119: Input parameters of xd_select_schema function....157
Table 120: Output parameters of xd_select_schema function....157
Table 121: Input parameters of xd_orbit_file_decimate function....159
Table 122: Output parameters of xd_orbit_file_decimate function....160
Table 123: Error messages of xd_orbit_file_decimate function....161
Table 124: Input parameters of xd_attitude_file_decimate function....162
Table 125: Output parameters of xd_attitude_file_decimate function....163
Table 126: Error messages of xd_attitude_file_decimate function....164
Table 127: Default schema's versions....165
Table 128: List of schema's versions....167
Table 129: Fixed Header Structure....173
Table 130: Fixed Header. Validity Period....174
Table 131: Fixed Header. Source....175
Table 132: Predicted Orbit File. Variable_Header....176
Table 133: Predicted Orbit File. Data_Block....177
Table 134: Precicted Orbit File. OSV....177
Table 135: Orbit Scenario File. Variable_Header....179
Table 136: Orbit Scenario File. Data_Block....180
Table 137: Orbit Scenario File. Orbit_Change....180
Table 138: Orbit Scenario File. Orbit....180
Table 139: Orbit Scenario File. Cycle....180
Table 140: MLST non linear terms....181
Table 141: MLST Harmonic term....181
Table 142 Orbit Scenario File. Time_of_ANX....181
Table 143: Orbit Scenario File. Data_Block....184
Table 144: Satellite Configuration File. Data Block....187
Table 145: Satellite Configuration File. NORAD_Data Structure....187
Table 146: Satellite Configuration File. Lib_Init Structure....187
Table 147: Satellite Configuration File. Orbit_InitStructure....187
Table 148: Satellite Configuration File. Low and Tight Tolerances Structure....188
Table 149: Attitude File. Data Block....190
Table 150: Attitude File. Attitude Angles Data....190
Table 151: Attitude File. Quaternions Data....191
Table 152: Attitude File. List of Attitude Angles....191
Table 153: Attitude File. List of Quaternions Data....191
Table 154: Attitude File. Attitude_Angles....191
Table 155: Attitude File. Quaternions....192
Table 156: Star Tracker Configuration File. Data Block....196
Table 157: Star Tracker Configuration File. Mispointing....196
Table 158: Star Tracker Configuration File. Star tracker limits....197
Table 159: Star Tracker Configuration File. Star_Trackers_Priority....197
Table 160: Star Tracker Configuration File. List_of_Star_Trackers....197

Table 161: Star Tracker Configuration File. Pre and Post Launch angles....197
Table 162: Star Tracker Configuration File. Rotation_Angles....197
Table 163: DEM Configuration File. Data Block....200
Table 164: DEM Configuration File. DEM Structure....200
Table 165: DEM Configuration File. DEM_User_Parameters Structure....200
Table 166: DEM Configuration File. DEM_Metadata Structure....201
Table 167: DEM Mini tile configuration. DEM_User_Parameters Structure....201
Table 168: Swath Definition File. Data Block....203
Table 169: Swath Definition File. Swath....203
Table 170: Swath Definition File. Refraction....203
Table 171: Swath Definition File. Swath Point....204
Table 172 Swath Definition File. Point Geometry....204
Table 173 Swath Definition File. Limb and Inertial Geometry....204
Table 174: Swath Definition File. Distance Point Geometry....205
Table 175: Swath Definition File. ASAR Geometry....205
Table 176: Swath Definition File. Satellite Nominal Attitude....205
Table 177: Swath Definition File. Satellite and Instrument Attitude....205
Table 178: Swath Definition File. Parameter Model....206
Table 179: Swath Definition File. Harmonic Model....206
Table 180: Swath Definition File. File Model....206
Table 181: Swath Definition File. Angle Model....207
Table 182: Swath Definition File. Matrix Model....207
Table 183: Swath Definition File. List_of_Parameters....207
Table 184: Swath Definition File. List_of_Harmonics_Pitch/Roll/Yaw....207
Table 185: Swath Definition File. Harmonic....208
Table 186: Swath Definition File. Offsets....208
Table 187: Swath Definition File. File....208
Table 188: Swath Definition File. Time Selection....208
Table 189: Swath Definition File. Time_Window....208
Table 190 Swath Definition File. Row....208
Table 191: Swath Definition File. Narrow Asar....209
Table 192: Swath Definition File. Wide Asar....209
Table 193: Swath Template File. Variable_Header....212
Table 194: Swath Template File. STF_Altitude....213
Table 195: Swath Template File. Orbit_Geometry....213
Table 196: Swath Template File. Orbit_State_Vector....213
Table 197: Swath Template File. Line_Altitude....213
Table 198: Swath Template File. Refraction....213
Table 199: Swath Template File. Data_Block....214
Table 200: Swath Template File. STF_Pt....214
Table 201: Swath Template File. Geodetic_Pt....214
Table 202: Swath Template File. Inertal_Pt....214
Table 203: Zone Database File. Data_Block....216
Table 204: Zone Database File. Zone....217
Table 205: Zone Database File. Polygon_Pt....217
Table 206: Station Database File. Data Block....220
Table 207: Station Database File. Location....220
Table 208: Station Database File. Spacecraft....221
Table 209: Station Database File. Mask_Point....221
Table 210: Precise Propagator Configuration File. Data Block....225
Table 211: Known problems....231

1. SCOPE

The EO_DATA_HANDLING Software User Manual provides:

- a detailed description of usage of the CFI functions included within the EO_DATA_HANDLING CFI software library.
- The format description of the Earth Observation Missions files as well as the available versions of those files.

2. ACRONYMS, NOMENCLATURE AND TERMINOLOGY

2.1.Acronyms

ANX	Ascending Node Crossing
AOCS	Attitude and Orbit Control Subsystem
ASCII	American Standard Code for Information Interchange
BOM	Beginning Of Mission
CFI	Customer Furnished Item
EOM	End Of Mission
ESA	European Space Agency
ESTEC	European Space Technology and Research Centre
GPL	GNU Public License
GPS	Global Positioning System
IERS	International Earth Rotation Service
I/F	Interface
LS	Leap Second
OBT	On-board Binary Time
OSF	Orbit Scenario File
SRAR	Satellite Relative Actual Reference
SUM	Software User Manual
TAI	International Atomic Time
UTC	Coordinated Universal Time
UT1	Universal Time UT1
WGS[84]	World Geodetic System 1984

2.2.Nomenclature

CFI	A group of CFI functions, and related software and documentation. that will be distributed by ESA to the users as an independent unit
CFI function	A single function within a CFI that can be called by the user
Library	A software library containing all the CFI functions included within a CFI plus the supporting functions used by those CFI functions (transparently to the user)

2.3.Note on Terminology

In order to keep compatibility with legacy CFI libraries, the Earth Observation Mission CFI Software makes use of terms that are linked with missions already or soon in the operational phase like the Earth Explorers.

This may be reflected in the rest of the document when examples of Mission CFI Software usage are proposed or description of Mission Files is given.

3. APPLICABLE AND REFERENCE DOCUMENTS

3.1.Applicable Documents

[GEN_SUM]	Earth Observation Mission CFI Software. General Software User Manual. EO-MA- DMS-GS-0002. Issue 4.4 05/07/12
[EE_FMT]	Earth Explorer File Format Standards. PE-TN-ESA-GS-0001 Issue 1.4 13/06/04
[GS_FMT]	Cryosat Ground Segment Mission Files Format Specification. CS-ID-ESA-GS-0224
[PDS_FMT]	Cryosat Ground Segment Payload Data Segment L0 Product Specification Format CS-ID-ACS-GS-0119

3.2.Reference Documents

[MCD]	Earth Observation Mission CFI Software. Mission Conventions Document. EO-MA- DMS-GS-0001. Issue 4.4 05/07/12.
[F_H_SUM]	Earth Observation Mission CFI Software. EO_FILE_HANDLING Software User Manual. EO-MA-DMS-GS-0008. Issue 4.4 05/07/12.
[LIB_SUM]	Earth Observation Mission CFI Software. EO_LIB Software User Manual. EO-MA-DMS-GS-0003. Issue 4.4 05/07/12.
[ORBIT_SUM]	Earth Observation Mission CFI Software. EO_ORBIT Software User Manual. EO-MA-DMS-GS-0004. Issue 4.4 05/07/12.
[POINT_SUM]	Earth Observation Mission CFI Software. EO_POINTING Software User Manual. EO-MA-DMS-GS-0005. Issue 4.4 05/07/12.
[VISIB_SUM]	Earth Observation Mission CFI Software. EO_VISIBILITY Software User Manual. EO-MA-DMS-GS-0006. Issue 4.4 05/07/12.
[IERS]	http://www.iers.org/iers/publications/bulletins/

4. INTRODUCTION

4.1.Functions Overview

This software library contains a set of functions for reading and writing Earth Observation Mission Files. The following CFI functions are included:

4.1.1.Reading routines

- **xd_read_fhr:** reads the fixed header for an Earth Observation XML file.
- **xd_read_bulletin:** reads the time correlations from an IERS bulletin B (1980 and 2010 format).
- **xd_read_bulletin_2:** reads the time correlations from a IERS bulletins A and B (only 2010 format).
- **xd_read_orbit_file**: reads orbit files consisting in a list of state vectors of the satellite in the orbit. The following files are supported: Predicted Orbit files, Restituted Orbit files and DORIS Preliminary files.
- **xd_read_doris**: reads DORIS Navigator files for CRYOSAT.
- **xd_read_doris_header:** reads the MPH and SPH data from a DORIS Navigator file for CRYOSAT.
- **xd_read_osf**: reads Orbit Scenario files.
- **xd_read_att:** reads a generic attitude file.
- **xd_read_star_tracker:** reads an star traker file for CRYOSAT.
- **xd_read_star_tracker_conf_file:** reads an star tracker configuration file for CRYOSAT.
- **xd_read_dem:** provides the points of a DEM that are adjacent to a given point.
- **xd_read_dem_config_file:** reads a DEM configuration file.
- **xd_read_sdf**: reads swath definition files.
- **xd_read_stf:** reads swath template files.
- **xd_read_stf_vhr:** reads the variable header for swath template files
- **xd_read_zone:** reads the parameters of one zone in a zone database file.
- **xd_read_zone_file:** reads a zone database file.
- **xd_read_zone_id:** reads the list of zone names from a zone database file.
- **xd_read_station:** reads the parameters of one station in a station database file.
- **xd_read_station_file:** reads a station database file.
- **xd_read_station_id:** reads the list of station names from a station database file
- **xd_read_star:** reads the parameters of one star in a star database file.
- **xd_read_star_file:** reads a star database file.
- **xd_read_star_id:** reads the list of star id. from a star database file
- **xd_read_tle:** reads a TLE file
- **xd_read_precise_propag_file:** reads a data file used to configure the numerical propagator

4.1.2.Writing routines

- **xd_write_orbit_file**: writes an orbit file using as input an structure with the data of the file
- **xd_write_osf**: writes an orbit scenario file using as input an structure with the data of the file
- **xd_write_doris:** writes a DORIS Navigator file.
- **xd_write_att:** writes a generic attitude file.
- **xd_write_stf**: writes a swath template file using as input the data structure containing the data for the swath.
- **xd_write_tle**: writes a TLE file using as input a data structure.

4.1.3.Functions to free memory

- **xd_free_orbit**: frees the memory allocated during the reading function **xd_read_orbit_file**.
- **xd_free_doris**: frees the memory allocated during the reading function **xd_read_doris**
- **xd_free_osf**: frees the memory allocated during the reading function **xd_read_osf**.
- **xd_free_sdf**: frees the memory allocated during the reading function **xd_read_sdf**.
- **xd_free_stf**: frees the memory allocated during the reading function **xd_read_stf**.
- **xd_free_stf_vhr**: frees the memory allocated during the reading function **xd_read_stf_vhr**.
- **xd_free_att**: frees the memory allocated during the reading function **xd_read_att**.
- **xd_free_star_tracker**: frees the memory allocated during the reading function
- xd_read_star_tracker.
- **xd_free_dem**: frees the memory allocated in the reading function **xd_read_dem**
- **xd_free_zone**: frees the memory allocated during the reading function **xd_read_zone**.
- **xd_free_zone_file**: rees the memory allocated during the reading function **xd_read_zone_file**.
- **xd_free_zone_id**: frees the memory allocated during the reading function **xd_read_zone_id**.
- **xd_free_station_file**: frees the memory allocated during the reading function
- xd_read_station_file.
- **xd_free_station_id**: frees the memory allocated during the reading function **xd_read_station_id**.

4.1.4.Validation of XML files

- **xd_xml_validate**: validates an XML file using an XML schema as reference.
- **xd_select_schema**: it returns the most recent schema name supported for a given file type and mission

4.2.Reading and writing files

When reading files, the user should be aware that:

- Many of the structures used for reading files contain dynamic data that is allocated within the reading function. In these cases, the memory has to be freed when it is not going to be used any more by calling the suitable function.
- The reading functions for each of the file types, does not read the fixed header. The fixed header could be read independently using the CFI function **xd_read_fhr**.
- When reading the fixed header with **xd_read_fhr**, the schema name is not read (the “schema” element in the output structure **xd_fhr** will be set to “_NOSCHEMA_”) . If required, the schema name and version should be read independently with the CFI functions in explorer_file_handling.

When writing files, the user should be aware that:

- The schema name and version can be written in the file in the following ways:
  - Setting the schema name in the “schema” element in the **xd_fhr** structure. When calling the **xd_write_*xxx*** function, the schema name and version will be written in the file. Note that if the schema name is set to “_NOSCHEMA_”, the schema attributes will no be written in the file.
  - After writing the file, by calling the function **xf_set_schema** (in explorer_file_handling). Note that the CFI function **xd_select_schema** allows to get the default schema name with which the file to be written is compliant.

4.3. Memory usage

Note: due to the implementation of the third-party library handling XML files, large amount memory may be needed by an application handling (reading or writing) a file with many entries. Therefore the user is recommended to perform handling of large files on computers equipped with adequate memory resources. If these resources are not available, the user has to properly configure virtual memory and take into account long execution times. In extreme cases, due to platform limitation or operating system settings, the operation may fail. In order to give an indication, a restituted orbit file covering a period of 10 days and a time interval between OSVs of 30 sec contains 28800 OSVs and its size on disk is about 14MB. The memory usage peak during the writing of such file is about 215MB.

5. LIBRARY INSTALLATION

For a detailed description of the installation of any CFI library, please refer to [GEN_SUM].

6. LIBRARY USAGE

Note that to use the EO_DATA_HANDLING software library, the following other CFI software libraries are required:

- EO_FILE_HANDLING (See [F_H_SUM]).

It is needed to have properly installed in the system the following external libraries:

- POSIX thread library:libpthread.so (pthread.lib for WINDOWS, with license LGPL)

To use the EO_DATA_HANDLING software library in a user application, that application must include in its source code:

- explorer_data_handling.h (for a C application)

To link correctly this application, the user must include in his linking command flags like (assuming *cfi_lib_dir* and *cfi_include_dir* are the directories where respectively all CFI libraries and include files have been installed, see [GEN_SUM] for installation procedures):

- LINUX:

  -I*cfi_include_dir* -L*cfi_lib_dir* -lexplorer_data_handling

  -lexplorer_file_handling –lpthread
- WINDOWS:

  /I *"cfi_include_dir"* /libpath:*"cfi_lib_dir"* libexplorer_data_handling.lib libexplorer_file_handling.lib pthread.lib
- MacOS:

  -I*cfi_include_dir* -L*cfi_lib_dir* -lexplorer_data_handling

  -lexplorer_file_handling -lpthread

All functions described in this document have a name starting with the prefix `xd_`

To avoid problems in linking a user application with the `EO_DATA_HANDLING` software library due to the existence of names multiple defined, the user application should avoid naming any global software item beginning with either the prefix `XD_` or `xd_`.

Finally, in order to use the function **xd_xml_validate** or its equivalent executable program **xml_validator**, it is neccessary the xerces-c 3.1.1 (or later) dynamic library and the SAX2Count executable program. Both of them are distributed under Apache 2.0 Licence (More information about xerces-c can be found in: http://xml.apache.org/xerces-c/index.html).

Both, **xd_xml_validate** and **xml_validator** internally calls the SAX2Count program, so that in order to find it, the path for SAX2Count has to be included in the environment variable PATH for all the operating systems; the xerces-c dynamic library must be also accesible, so its path must be included in the corresponding environment variable:

- For LINUX, in the LD_LIBRARY_PATH variable.
- For WINDOWS, in the PATH variable.
- For MacOS, in the DYLD_LIBRARY_PATH variable.

It is possible to call the following CFI functions from a user application.

Table 1: CFI functions included within EO_DATA_HANDLING library

Function Name	Enumeration value	Long
Main CFI Functions		
xd_read_fhr	XD_READ_FHR_ID	0
xd_read_bulletin	XD_READ_BULLETIN_ID	1
xd_read_orbit_file	XD_READ_ORBIT_FILE_ID	2
xd_read_doris	XD_READ_DORIS_ID	3
xd_read_doris_header	XD_READ_DORIS_HEADER_ID	4
xd_read_osf	XD_READ_OSF_ID	5
xd_read_sdf	XD_READ_SDF_ID	6
xd_read_stf	XD_READ_STF_ID	7
xd_read_stf_vhr	XD_READ_STF_VHR_ID	8
xd_read_att	XD_READ_ATT	9
xd_read_star_tracker	XD_READ_STAR_TRACKER_ID	10
xd_read_str_conf_file	XD_READ_STR_CONF_FILE_ID	11
xd_read_dem_config_file	XD_READ_DEM_CONFIG_FILE_ID	12
xd_read_dem	XD_READ_DEM_ID	13
xd_read_star	XD_READ_STAR_ID	14
xd_read_star_file	XD_READ_STAR_FILE_ID	15
xd_read_star_id	XD_READ_STAR_ID_ID	16
xd_read_station	XD_READ_STATION_ID	17
xd_read_station_file	XD_READ_STATION_FILE_ID	18
xd_read_station_id	XD_READ_STATION_ID_ID	19
xd_read_zone	XD_READ_ZONE_ID	20
xd_read_zone_file	XD_READ_ZONE_FILE_ID	21
xd_read_zone_id	XD_READ_ZONE_ID_ID	22
xd_write_orbit_file	XD_WRITE_ORBIT_FILE_ID	23
xd_write_doris	XD_WRITE_DORIS_ID	24
xd_write_osf	XD_WRITE_OSF_ID	25
xd_write_stf	XD_WRITE_STF_ID	26
xd_write_att	XD_WRITE_ATT_ID	27
xd_xml_validate	XD_XML_VALIDATE_ID	28

Function Name	Enumeration value	Long
xd_read_tle	XD_READ_TLE	29
xd_write_tle	XD_WRITE_TLE	30
xd_read_precise_propag_file	XD_READ_PRECISE_PROPAG_FILE_ID	31
Error Handling Functions		
xd_verbose	not applicable	
xd_silent		
xd_get_code		
xd_get_msg		
xd_print_msg		

Notes about the table:

- To transform the extended status flag returned by a CFI function to either a list of error codes or a list of error messages, the enumeration value (or the corresponding long value) described in the table must be used
- The error handling functions have no enumerated values

Whenever available it is strongly recommended to use enumeration values rather than integer values.

6.1.Usage hints

Every CFI function has a different length of the Error Vector, used in the calling I/F examples of this SUM and defined at the beginning of the library header file. In order to provide the user with a single value that could be used as Error Vector length for every function, a generic value has been defined (XD_ERR_VECTOR_MAX_LENGTH) as the maximum of all the Error Vector lengths. This value can therefore be safely used for every call of functions of this library.

6.2.General Enumerations

The aim of the current section is to present the enumeration values that can be used rather than integer parameters for some of the input parameters of the EO_DATA_HANDLING routines, as shown in the table below. The enumerations presented in [GEN_SUM] are also applicable.

Table 2: Enumerations within EO_DATA_HANDLING library

Input	Description	Enumeration value	Long
Boolean values	False value	XD_FALSE	0
	True value	XD_TRUE	1
Returned status code	Error	XD_ERR	-1
	Ok status	XD_OK	0
	Warning	XD_WARN	1

Input	Description	Enumeration value	Long
Time initialization	Select the whole file	XD_SEL_FILE	0
	Select a time range	XD_SEL_TIME	1
	Select an orbit range	XD_SEL_ORBIT	2
	Select the default value	XD_SEL_DEFAULT	3
Time reference	Undefined	XD_TIME_UNDEF	-1
	TAI	XD_TIME_TAI	0
	UTC	XD_TIME_UTC	1
	UT1	XD_TIME_UT1	2
	GPS	XD_TIME_GPS	3
Attitude data type	Quaternions	XD_ATT_QUATERNIONS	0
	Angles	XD_ATT_ANGLES	1
Ray tracing model		XD_NO_REF	0
		XD_STD_REF	1
		XD_USER_REF	2
		XD_PRED_REF	3
		XD_STD_REF_N	10
		XD_USER_REF_N	20
		XD_PRED_REF_N	30
		XD_US76_REF	300
		XD_TROPIC_REF	301
		XD_MID_SUM_REF	302
		XD_MID_WIN_REF	303
		XD_SUBAR_SUM_REF	304
		XD_SUBAR_WIN_REF	305
		XD_LUT_REF	400
		XD_US76_REF_N	3000
		XD_TROPIC_REF_N	3001
		XD_MID_SUM_REF_N	3002
		XD_MID_WIN_REF_N	3003
		XD_SUBAR_SUM_REF_N	3004
		XD_SUBAR_WIN_REF_N	3005
		XD_LUT_REF_N	4000
Swath Types		XD_OPEN_SWATH	0
		XD_CLOSED_SWATH	1
Swath Point types		XD_GEODETIC_SWATH_TYPE	0
		XD_INERTIAL_SWATH_TYPE	1
Swath geometry definition = algorithm		XD_SWATH_POINTING_GEOM	0
		XD_SWATH_DISTANCE_GEOM	1
		XD_SWATH_LIMB_GEOM	2
		XD_SWATH_INERTIAL_GEOM	3
		XD_SWATH_SUBSATELLITE_GEOM	4
		XD_SWATH_ASAR_GEOM	5
Asar swath types		XD_NO_ASAR	0
		XD_NARROW_ASAR	1
		XD_WIDE_ASAR	2

Input	Description	Enumeration value	Long
Orbit file types	Orbit Scenario File	XD_REF_FILETYPE_OSF	0
	Orbit Event file used as an OSF	XD_REF_FILETYPE_OEF_OSF	1
	FOS Predicted Orbit File	XD_REF_FILETYPE_POF	2
	Orbit Event file used as a POF	XD_REF_FILETYPE_OEF_POF	3
	DORIS Navigator File	XD_REF_FILETYPE_DORIS_NAV	4
	FOS Restituted Orbit File	XD_REF_FILETYPE_ROF	5
	DORIS Preliminary Orbit File	XD_REF_FILETYPE_DORIS_PREM	6
	DORIS Precise Orbit File	XD_REF_FILETYPE_DORIS_PREC	7
Orbit modes and file types	Unknown	XD_UNKNOWN_TYPE	0
	Detect automatically	XD_AUTO	1
	Orbit from orbital change info	XD_ORBIT_CHANGE	2
	Orbit from one state vector	XD_STATE_VECTOR	3
	Orbit Scenario File	XD_OSF_TYPE	4
	FOS Predicted Orbit File	XD_POF_TYPE	5
	FOS Restituted Orbit File DORIS Preliminary Orbit File DORIS Precise Orbit File	XD_ROF_TYPE	6
	DORIS Navigator File	XD_DORIS_TYPE	7
	Predicted orbit file plus DORIS Navigator file	XD_POF_N_DORIS_TYPE	8
	Orbit Event file used as an OSF	XD_OEF_OSF_TYPE	9
	Orbit Event file used as a POF	XD_OEF_POF_TYPE	10
	IERS Bulletin B file	XD_IERS_B_TYPE	11
	Two line elements file	XD_TLE_TYPE	12
	Swath Template file	XD_STF_TYPE	13
	DORIS Precise file	XD_DORISPREC_TYPE	14
	Doris Preliminary file	XD_DORISPREM_TYPE	15
	Attitude file	XD_ATT_TYPE	16
	Swath Control file	XD_SCF_TYPE	17
	Precise Propagation configuration file	XD_PRECISE_PROPAG_TYPE	18
	DEM Configuration file	XD_DEMCFG_TYPE	19
	Satellite Configuration file	XD_SATCFG_TYPE	20
	Ground Station Database file	XD_GND_DB_TYPE	21
	Swath Definition file	XD_SW_DEF_TYPE	22
	Zone Database file	XD_ZON_DB_TYPE	23
	Star Tracker file	XD_STR1ATT_TYPE	24
	IERS Bulletin A file	XD_IERS_A_TYPE	25
	IERS Bulletin B plus A	XD_IERS_B_AND_A_TYPE	26

Input	Description	Enumeration value	Long
Coordinate systems	Barycentric Mean of 2000.0	XD_BAR_MEAN_2000	1
	Heliocentric Mean of 2000.0	XD_HEL_MEAN_2000	2
	Geocentric Mean of 2000.0	XD_GEO_MEAN_2000	3
	Mean of date	XD_MEAN_DATE	4
	True of date	XD_TRUE_DATE	5
	Earth Fixed	XD_EARTH_FIXED	6
	Barycentric Mean of 1950.0	XD_BAR_MEAN_1950	7
	Galactic	XD_GALACTIC	8
	Satellite relative actual reference	XD_SAT_ACT_REF	9
	Quasi-Mean of Date	XD_QUASI_MEAN_DATE	10
	Pseudo-True of Date	XD_PSE_TRUE_DATE	11
	Pseudo Earth Fixed	XD_PSEUDO_EARTH_FIXED	12
	Topocentric	XD_TOPOCENTRIC	13
	Satellite reference	XD_SAT_REF	14
	Satellite relative reference	XD_SAT_REL_REF	15
Attitude reference frames	Orbital reference frame	XD_SAT_ORBITAL_REF	0
	Satellite nominal attitude frame	XD_SAT_NOMINAL_ATT	1
	Satellite attitude frame	XD_SAT_ATT	2
	Instrument attitude frame	XD_INSTR_ATT	3
Different models for DEM	ACE Model (deprecated)	XD_DEM_ACE_MODEL	0
	GETASSE 30 v1	XD_DEM_GETASSE30_V1	1
	GETASSE 30 v2	XD_DEM_GETASSE30_V2	2
	ACE2 9 seconds	XD_DEM_ACE2_9SEC	3
Zone types	zone is not defined as an input and must be read from a file	XD_NOT_DEFINED	-1
	Point zone	XD_POINT	0
	Circular zone	XD_CIRCLE	1
	Segment zone	XD_SEGMENT	2
	Polygonal zone	XD_POLYGON	3
Projection types	Read projection from DB file	XD_READ_DB	0
	Use gnomonic projection	XD_GNOMONIC	1
	Use rectangular projection	XD_RECTANGULAR	2
Validation Status	Invalid file	XD_XML_INVALID	-1
	Valid file	XD_XML_VALID	0
Quality Index	Adjusted out of orbit manoeuvre period	XD_3_ADJUST_NOMI	1
	Adjusted during an orbit manoeuvre	XD_4_ADJUST_DMAN	2
	Interpolated during a data gap	XD_5_INTERP_DGAP	3
	Extrapolated from less than 1 day	XD_6_EXTRAP_LT1D	4
	Extrapolated from more than 1 day, but less than 2 days	XD_7_EXTRAP_1D2D	5
	Extrapolated from more than 2 days	XD_8_EXTRAP_GT2D	6
	Extrapolated after an orbit manoeuvre	XD_8_EXTRAP_AMAN	7
Draw modes for the SCF	SOLID	XD_SCF_DRAW_SOLID	0
	DASHED	XD_SCF_DRAW_DASHED	1
	DOTTED	XD_SCF_DRAW_DOTTED	2
	TIMELINE	XD_SCF_DRAW_TIMELINE	3
Fill modes for the SCF	SOLID	XD_SCF_FILL_SOLID	0
	HOLLOW	XD_SCF_FILL_HOLLOW	1
Reference time values	TAI reference	XD_TIME_REF_OF_TAI	0
	UTC reference	XD_TIME_REF_OF_UTC	1
	UT1 reference	XD_TIME_REF_OF_UT1	2

Input	Description	Enumeration value	Long
DEM Data Source Types for GETASSE30 V1 and V2	Data from ACE (land-ice/snow)	XD_DEM_GETASSE30_SOURCE_ACE	0
	Data from MSS (Sea)	XD_DEM_GETASSE30_SOURCE_MSS	1
	Data from EGM96 (Sea-Ice)	XD_DEM_GETASSE30_SOURCE_EGM96	2
	Data from SRTM30 (Land)	XD_DEM_GETASSE30_SOURCE_SRTM30	3
DEM Data Source Types for ACE2 9secs	Pure SRTM (above 60°N pure GLOBE data, below 60S pure ACE [original] data)	XD_DEM_ACE2_SOURCE_SRTM0	0
	SRTM voids filled by interpolation and/or altimeter data	XD_DEM_ACE2_SOURCE_SRTM1	1
	SRTM data warped using the ERS-1 Geodetic Mission	XD_DEM_ACE2_SOURCE_SRTM2	2
	SRTM data warped using EnviSat & ERS-2 data	XD_DEM_ACE2_SOURCE_SRTM3	3
	Mean lake level data derived from Altimetry	XD_DEM_ACE2_SOURCE_SRTM_LAKE	4
	GLOBE/ACE data warped using combined altimetry (only above 60°N)	XD_DEM_ACE2_SOURCE_SRTM_GLOBE	5
	Pure altimetry data (derived from ERS-1 Geodetic Mission, ERS-2 and EnviSat data using Delaunay Triangulation and Bilinear interpolation)	XD_DEM_ACE2_SOURCE_SRTM_ALT	6
IERS Bulletin type	Bulletin A	XD_BULLETIN_A	0
	Bulletin B	XD_BULLETIN_B	1
Data file type for xd_eocfi_file structure	Orbit file type (POF or ROF)	XD_ORBIT_FILE	0
	Orbit Scenario file	XD_OSF_FILE	1
	DORIS Navigator file	XD_DORIS_FILE	2
	IERS Bulletin file	XD_BULLETIN_FILE	3
DEM cache type	Computations performed without memory cache	XD_NO_CACHE	0
	Computations performed with preloadmemory cache	XD_PRELOAD_CACHE	1
	Computations performed with FIFO memory cache	XD_FIFO_CACHE	2

The use of the previous enumeration values could be restricted by the particular usage within the different CFI functions. The actual range to be used is indicated within a dedicated reference named ***allowed range***. When there are not restrictions to be mentioned, the allowed range column is populated with the label ***complete***.

6.3.Data Structures

The aim of this section is to present the data structures that are used in the EO_DATA_HANDLING library. These structures are used as output/inputs in the reading/writing routines. The following table show the data structures with their names and the data that contains:

Table 3: EO_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
xd_fhr	Fixed header data	file_name	char [XD_MAX_STR]	File name
		schema	char [XD_MAX_STR]	Schema file
		file_description	char [XD_MAX_STR]	File description
		mission	char [XD_MAX_STR]	Mission name
		file_class	char [XD_MAX_STR]	File class
		file_type	char [XD_MAX_STR]	File type
		version	long	File version
		val_start_date	char [32]	Validity start date
		val_stop_date	char [32]	Validity stop date
		system	char [XD_MAX_STR]	System name
		creator	char [XD_MAX_STR]	Creator name
		creator_version	char [XD_MAX_STR]	Creator version
		creation_date	char [32]	Creation date
xd_fileinfo	File info data for getting the default schema	sat_id	long	“Satellite ID” enumeration value (see [GEN_SUM])
		filetype	XD_File_types	File type (see enumeration in Table 2)
xd_bulb_table	Data for one entry read from a IERS bulletin	day	double	MJ200 UTC Time
		ut1_utc	double	Difference between UT1 and UTC
		ut1_tai	double	Difference between UT1and TAI
xd_iers_bulletin_b	Data for time corre lations read from a IERS bulletin	table1	xd_bulb_table[100]	First table data in the IERS bulletin
		table2	xd_bulb_table[100]	ble Difference between UT1and TAI

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
xd_eocfi_file_union	Union containin any of the following data structures	orbit_file	xd_orbit_file	Data from an orbit file (POF or ROF)
		osf_file	xd_osf_file	Data from an Orbit Scenario File
		doris_file	xd_doris_file	Data from a DORIS Navigator File
		bulletin_file	xd_iers_bulletin_file	Data from an IERS bulletin file (A or B)
xd_eocfi_file	Data from an EOCFI file (Orbit file, OSF, DORIS Navigator or IERS file)	file_type	long	File type (according to XD_data_file_type_enum)
		eocfi_file	xd_eocfi_file_union	File data
xd_eocfi_file_set	Set of EOCFI files	num_files	long	Number of structures with the data from the files
		eocfi_file_array	xd_eocfi_file*	Array with the data structures
xd_polar_motion_params	Polar motion parameters read from IERS bulletins	x	double	x-axis is in the direction of the IERS Reference Meridian (IRM),
		y	double	y-axis is in the direction 90 degrees West longitude
xd_iers_bulletin_b_rec	Data for one entry read from a IERS bulletin B	day	double	MJD200 UTC time
		ut1_utc	double	Difference between UT1 and UTC
		ut1_tai	double	Difference between UT1 and TAI
		polar_motion_params	xd_polar_motion_params	Polar motion parameters
xd_iers_bulletin_a_rec	Data for one entry read from a IERS bulletin A	day	double	MJD200 UTC time
		ut1_utc	double	Difference between UT1 and UTC
		ut1_tai	double	Difference between UT1 and TAI
		polar_motion_params	xd_polar_motion_params	Polar motion parameters

Structure name	Description	Structure Data		
		Variable Name	C type	Description
xd_polar_motion_formula	Polar motion prediction formula paramters	ax	double	x parameter formula: constant term
		bx	double	x parameter formula: cos(A) coefficient
		cx	double	x parameter formula: sin(A) coefficient
		dx	double	x parameter formula: cos(C) coefficient
		ex	double	x parameter formula: sin(C) coefficient
		ay	double	y parameter formula: constant term
		by	double	y parameter formula: cos(A) coefficient
		cy	double	y arameter formula: sin(A) coefficient
		dy	double	y parameter formula: cos(C) coefficient
		ey	double	y parameter formula: sin(C) coefficient formula:
		A_ref	double	Reference day for A parameter formula
		A_div	double	Divisor for A parameter formula
		C_ref	double	Reference day for C parameter formula
		C_div	double	Divisor for C parameter formula
xd_time_correlation_formula	It contains the parameters for the UT1-UTC prediction formula	a	double	Constant parameter in formula
		b	double	Linear parameter in formula
		b_ref	double	Reference orbit in formula
xd_iers_bulletin_b_file	It contains values read from an IERS Bulletin B file	bulletin_id	char[]	Bulletin date and issue
		num_final_table	long	Number of record in final table
		num_preliminary_table	long	Number of records in preliminary table
		num_smoothed_table	long	Number of records in smoothed table
		final_table	xd_iers_bulletin_b_rec	Data read from Final table
		preliminary_table	xd_iers_bulletin_b_rec	Data read from Preliminary table
		smoothed_table	xd_iers_bulletin_b_rec	Data read from Smoothed table
xd_iers_bulletin_a_file	It contains values read from an IERS Bulletin A file	bulletin_id	char[]	Bulletin date and issue
		num_rec_pred_table	long	Number of record in Prediction table
		prediction_table	xd_iers_bulletin_b_r	Data read from Prediction table

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
			ec	
		polar_motion_formula	xd_polar_motion_formula	Parameters read for Polar motion formula
		time_correlation_formula	xd_time_correlation_formula	Parameters read for time correlation formula
xd_iers_bulletin_file_union (union C type)	It contains the values read from Bulletin A or Bulletin B (only one of them)	iers_bulletin_a_file	xd_iers_bulletin_a_file	Bulletin A data
		iers_bulletin_b_file	xd_iers_bulletin_b_file	Bulletin B data
xd_iers_bulletin_file	Bulletin type and Bulletin data	bulletin_type	long	It can take the following values: –XD_BULLETIN_A –XD_BULLETIN_B
		iers_bulletin_file	xd_iers_bulletin_file_union	Bulletin data union
xd_time_rec	It contains the time correlations for a given time	tai_time	double	TAI time
		ut1_time	double	UT1 time
		tai_utc	double	Difference between TAI and UTC time
		tai_ut1	double	Difference between TAI and UT1 time
		tai_gps	double	Difference between TAI and GPS time
xd_osv_rec	It contains a satellite state vector for a given time	tai_time	double	TAI time for the state vector
		utc_time	double	UTC time for the state vector
		ut1_time	double	UT1 time for the state vector
		abs_orbit	double	Absolute orbit
		ref_frame	long	Reference frame
		time_ref_of	long	Reference time to be considered as base. This value is related to Time_Reference tag in orbit file. This parameter takes the values given by enumeration *Reference time values* (see Table 1). For more details on this field see section 7.5.1 of [ORBIT_SUM].
		pos	double[3]	Position vector (x, y, z components)
		vel	double[3]	Velocity vector (x, y, z components)
		quality	double	Quality index .For DORIS Preliminary and DORIS Precise Orbit files, this value corresponds with the

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
				enumeration “Quality Index” (See Table 2)
xd_orbit_file	Structure for storing the data read from an orbit file	num_rec	long	Number of records
		osv_rec	xd_osv_rec*	Array with the state vectors
xd_doris_file	Structure for storing the data read from a DORIS Navigator file	num_rec	long	Number of records
		osv_rec	xd_osv_rec	State vectors array (EF)
		osv_rec_j2	xd_osv_rec	State vectors array (J2000)
		leap_time	double	Leap time
		leap_sign	int	Leap time sign
		abs_orbit	long	First absolute orbit number
		rel_orbit	long	First relative orbit number

Structure name	Description	Structure Data		
		Variable Name	C type	Description
xd_doris_mph_sph	Structure for the main and specific product headers	filename	char [XD_MAX_STR]	The description for these fields can be found in [PDS_FMT]
		sensing_start	char [30]	
		sensing_stop	char [30]	
		abs_orbit	long	
		delta_ut1	long	
		rel_orbit	long	
		leap_utc	char [XD_MAX_STR]	
		leap_sign	int	
		leap_err	int	
		num_dsd	long	
		ds_offset	long	
		num_dsr	long	
		proc_stage_code	char [5]	
		ref_doc	char [24]	
		proc_time	char [31]	
		software_version	char [15]	
		phase	char [2]	
		cycle	long	
		state_vector_time	char [31]	
		x_position	double	
		y_position	double	
		z_position	double	
		x_velocity	double	
		y_velocity	double	
		z_velocity	double	
		state_vector_source	char [3]	
		ascii_utc_time_before_leap	double	
		product_err	char [2]	
		tot_size	long	
		num_data_sets	long	

Structure name	Description	Structure Data		
		Variable Name	C type	Description
		sph_descriptor	char [29]	
		sensing_start_tai	char [31]	
		abs_orbit_start	long	
		rel_time_asc_node_start	double	
		sensing_stop_tai	char [31]	
		abs_orbit_stop	long	
		rel_time_asc_node_stop	double	
		equator_cross_time	char [31]	
		equator_cross_long	long	
		ascending_flag	char [2]	
		start_lat	long	
		start_long	long	
		stop_lat	long	
		stop_long	long	
		num_isps	long	
		num_missing_isps	long	
		num_error_isps	long	
		num_discarded_isps	long	
		num_rs_isps	long	
		num_rs_corrections	long	
		dsr_size	long	
xd_osf_rec	It contains the data for an orbital change in an orbit scenario file	abs_orb	long	Absolute orbit number
		rel_orb	long	Relative orbit number
		cycle_days	long	Cycle length in days
		cycle_orbits	long	Number of orbits in a cycle
		mlst	double	Mean local solar time (in hours)
		mlst_drift	double	Mean local solar time drift (seconds per day)
		inclination	double	Orbit inclination
		drift_mode	long	Flag for choosing between inclination of drift model
		anx_tai	double	ANX TAI time

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
		anx_ut1	double	ANX UT1 time
		anx_utc	double	ANX UTC time
		anx_long	double	ANX longitude
		cycle	long	Cycle number
		phase	long	Phase number
		time_ref_of	long	Reference time to be considered as base. This value is related to Time_Reference tag in orbit file. For OSF, this value is always XD_TIME_REF_OF_UT1 (see enumeration *Reference time values* in Table 1).
xd_osf_file	Structure for storing the data read from an orbit scenario file	num_rec	long	Number of records
		osf_rec	xd_osf_rec*	Array of state vectors
xd_swath_geometry	It contains the swath geometry	geom_type	long	Geometry type
		az	double[3]	Azimuth points
		el	double[3]	Elevation points
		alt	double[3]	Altitude points
		distance	double[3]	Distance
xd_harmonic_data		num_terms	long[3]	Number of harmonics coefficient(pitch, roll and yaw)
		harmonic_type_pitch	long[XD_MAX_NU M	Harmonic type
		harmonic_type_roll	long[XD_MAX_NU M	Harmonic type
		harmonic_type_yaw	long[XD_MAX_NU M	Harmonic type
		harmonic_coef_pitch	double [XD_MAX_NUM_HA RMONIC]	Harmonic coefficient
		harmonic_coef_roll	double [XD_MAX_NUM_HA RMONIC]	Harmonic coefficient
		harmonic_coef_yaw	double [XD_MAX_NUM_HA RMONIC]	Harmonic coefficient
xd_param_model_str		model	long	Model type
		param_num	long	Number of parameters
		model_param	double [XD_NUM_MODEL_ PARAM]	Model Parameters
xd_harmonic_		angle_type	long	Angle type

Structure name	Description	Structure Data		
		Variable Name	C type	Description
model_str		harmonics	xd_harmonic_data	Harmonic data
		offsets	double [3]	Offsets
xd_file_model_str		num_files	long	Number of files
		files	char **	file list
		aux_file	char *	Auxiliary file
		time_ref	long	Time reference
		time0	double	Start time
		time1	double	Stop time
xd_angle_model_str		angles	double [3]	angles
		offsets	double [3]	offsets
xd_matrix_model_str	Matrix model	att_matrix	double [3][3]	Attitude matrix model
		offsets	double [3]	Offsets
xd_attitutude_model_str	Attitude model structure	attitude_model	long	Attitude model type
		data	Attitude union data	Attitude union. One of the attitude structures.
Attitude union data	One of the following attitude structures	AOCS	long	AOCS model
		param_mode	xd_param_model_st	Parameters model
		harmonic_mode	xd_harmonic_model_str	Harmonic model
		file_mode	xd_file_model_str	File model
		angle_mode	xd_angle_model_str	Angle Model
		matrix_mode	xd_matrix_model_str	Matrix Model
xd_asar_geometry	ASAR geometry	asar_type	long	ASAR Swath types
		slant_range_left	double	Parameter for narrow and wide ASAR
		slant_range_righ t	double	Parameter only for wide ASAR
xd_sdf_rec	Swath Definition data	swath_descr	char [XD_MAX_STR]	Swath description
		swath_id	char [XD_MAX_STR]	Swath_id
		swath_type	long	Swath type (XD_Swath_type_enum)
		num_swath_rec	long	Number of swath records to write in a single OEF
		refr_mode	long	Refraction mode (XD_Target_ray_enum)
		freq	double	Frequency (Hz)
		num_points	long	Number of points in the instantaneous swath

Structure name	Description	Structure Data		
		Variable Name	C type	Description
		swath_geom	xd_swath_geometry *	Swath geometry
		asar_geom	xd_asar_geometry	ASAR parameters
		sat_nom_att	xd_attitude_model_s	Attitude data for sat. nominal att
		sat_att	xd_attitude_model_s	Attitude data for sat. attribute
		instr_att	xd_attitude_model_str *	Attitude data for instrument att
xd_sdf_file	Swath definition file data	num_rec	long	Number of swath records in a SDF
		sdf_rec	xd_sdf_rec *	Swath record data array
xd_stf_pt	Swath point definition structure	lon	double	Longitude or RA
		lat	double	Latitude or Dec
xd_stf_rec	Swath template record data	num_points	long	Number of points in the instantaneous swath
		stf_pt	xd_stf_pt*	Array with the points of the instantaneous swath
xd_stf_vhr	Swath template variable header data	stf_name	char *	swath template file name
		sdf_name	char [XF_MAX_PATH_LE NGTH]	Reference swath definition file
		swath_type	XD_Swath_type_enum	Swath type
		swath_point_type	XD_Swath_point_type_enum	Swath point type
		time_step	double	
		refr_mode	long	Refraction model
		freq	double	Frequency (Hz)
		num_points	long	Number of points in the instantaneous swath
		altitude	double*	Array with the values of the altitudes of the points
		geom_flag	long	true if the geometry of the orbit is defined. False if the OSV
		rep_cycl	long	repeat cycle
		cycle_length	long	cycle length
		mlst_drift	double	MLST drift
		abs_orbit	long	Absolut orbit
		pos	double [3]	ANX position vector

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
		vel	double [3]	ANX velocity vector
xd_stf_file	Swath template file data	num_rec	long	number of points in the swath
		vhr	xd_stf_vhr	variable header
		stf_rec	xd_stf_rec *	array with the points in the swath
xd_att_rec	Attitude record	time_ref	long	Time reference
		time	double	time (MJD2000)
		data	double [4]	Quaternions or angles. For angles, the fourth value is dummy
xd_att_file	Attitude file data	sat_ref	long	target reference frame
		source_ref	long	initial reference frame: Inertial reference frame
		data_type	long	angles or quaternions (see XD_Attitude_data_type_enum)
		num_rec	long	number of records in the attitude lists
		max_gap	double	Maximum time gap between two consequtive records
		att_rec	xd_att_rec*	array with the angle/ quaternion records
xd_tracker_limits	star trackers limits data	max_penalty	double	Maximum penalty for the quaternions
		norm_thr	double	Threshold for the modulus of the quaternion
		max_gap	double	Maximum time gap between two consequtive quaternions
xd_tracker_conf_file	star trackers cofiguration file data	aberr_correction	long	Aberration correction flag: -1 = Aberration correction with transposed matrix 0 = No aberration 1 = Aberration correction
		satellite	char [XD_MAX_STR]	Satellite name
		str_limit	xd_tracker_limits	Star tracker limits
		str_att_rot	double [3][3]	Satellite Attitude to star tracker frame rotation matrix
xd_star_tracker	Star tracker record	quaternion	float[4]	Quaternions
		time	double	MJ2000 in TAI
		status	unsigned char	quaternion status
xd_star_tracker_file	star tracker file data	str_id	long	Star tracker Id (1,2 or 3)
		num_rec	long	number of lines
		str_rec	xd_star_tracker*	array with the star tracker

Structure name	Description	Structure Data		
		Variable Name	C type	Description
				records
xd_dem_ace	DEM configuration data for ACE model (deprecated)	dir	char[100]	Directory where the DEM files are stored
		res_X	double	Interval between points along X-axis
		res_Y	double	Interval between points along Y-axis
		res_unit	double	Conversion factor from x,y units to the res_X, res_Y units. For example, if res_X is given in seconds and X in degrees then res_unit=3600
		X_num_points	long	Number of points along X-axis (columns)
		Y_num_points	long	Number of points along Y-axis (files)
		x_range	double	longitude of the X-axis for one file (grid).
		x_range	double	longitude of the Y-axis for one file (grid).
		data_size	long	Size in bytes of the data stored in the files
		data_type	long	data type (int, long, float, double)
		north_alt	double[4]	Altitude at the North pole cell
		south_alt	double[4]	Altitude at the South pole cell
		offset_x	double	Distance from the middle of a cell to the vertical side.
		offset_y	double	Distance from the middle of a cell to the horizontal side.
xd_dem_mini_tiles	Mini-tile configuration parameters for maximum altitude DEM algorithm	file_name	char[XD_MAX_STR]	Name of the maximum altitude file
		lon_size	double	Mini-tile longitude size [degrees]
		lat_size	double	Mini-tile latitude size [degrees]
xd_dem_user_params	User configuration parameters for DEM	directory	char[XD_MAX_STR]	Directory where the DEM files are stored
		cache_type	long	Cache type (DEM cache type enumeration)
		cache_max_size	long	Cache maximum size (in MegaBytes)
		mini_tiles	xd_dem_mini_tiles	DEM mini-tile configuration for maximum altitude algorithm
		geoid_computation	long	Flag to indicate if geoid

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
				computation must be performed or not (see DEM geoid flag enum)
		geoid_num_harmonics	long	Number of harmonics to be used in geoid computation
xd_dem_metadata	DEM metadata	model	long	DEM Model
xd_dem_config_file	DEM configuration data	model	long	DEM model
		dem_data	xd_dem_ace *	DEM ACE data (deprecated)
		dem_user_params	xd_dem_user_params	User configuration parameters
		dem_metadata	xd_dem_metadata	DEM extra information
xd_dem_point	DEM file point	lon	double	longitude
		lat	double	latitude
		alt	double	altitude
xd_dem_file	DEM file	num_points_X	long	Number of points along the longitude
		num_points_Y	long	Number of points along the latitude
		point	xd_dem_point**	DEM points
xd_star_rec	Star data	flag	long	True if the star was found in the star database file.
		star_id	char [XD_MAX_STR]	Star ID
		par	double	Parallax of the star at JD2000 (rads)
		mu_ra	double	RA's proper motion at JD2000 (rad/century)
		mu_dec	double	DEC's proper motion at JD2000 (rad/century)
		rad_vel	double	Radial velocity of the star (km/s)
		star_ra	double	RA of the star at JD2000 (rads)
		star_dec	double	DEC of the star at JD2000 (rads)
xd_star_file	Structure containing all relevant information contained in the star's database file	num_rec	long	Number of stars
		star_rec	xd_star_rec *	Array with all the star data
xd_station_rec	Station record data	station_id	char [XD_MAX_STR]	Station ID
		descriptor	char [XD_MAX_STR]	Description of the station

Structure name	Description	Structure Data		
		Variable Name	C type	Description
		antenna	char [XD_MAX_STR]	Describes the frequency band in which the anntena works.
		purpose	char [XD_MAX_STR]	Purpose
		type	char [XD_MAX_STR]	Not used.
		num_mask_pt	long	Number of points to define the antenna
		azimuth	double [XD_VERTICES]	Azimuth and elevation defining the antenna mask.
		elevation	double [XD_VERTICES]	
		station_long	double	Station longitude
		station_lat	double	Station latitude
		station_alt	double	Station altitude
		proj_long	double [XD_VERTICES]	longitude/latitude points for the station zone that are equivalent to the set of azimuth/elevation points. The longitude/latitude points are not read from the file but computed in xv_station_vis_time.
		proj_lat	double [XD_VERTICES]	
		points	long	Number of points in the azimuth/elevation and in proj_long/proj_lat arrays.
		long_max	double	Maximum longitude of the station zone
		lat_max	double	Maximum latitude of the station zone
		long_min	double	Minimum longitude of the station zone
		lat_min	double	Minimum latitude of the station zone
		mission_list	long	Number of spacecrafts defined for the station
		mission_name	char[XD_MISSIONS][XD_MAX_STR]	Names of the spacecrafts defined for the station
		mis_aos_el	double[XD_MISSIONS]	Elevations for acquisition of signal to defined spacecrafts
		mis_los_el	double[XD_MISSIONS]	Elevations for loss of signal to the defined spacecrafts
		mask_type	char[XD_MISSIONS][XD_MAX_STR]	Mask type for the spacecrafts defined in the station

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
xd_station_file		num_rec	long	Number of stations
		station_rec	xd_station_rec *	Array of station records
xd_zone_point	Longitude and latitude point	pt_long	double	Longitude
		pt_lat	double	Latitude
xd_zone_rec	Zone record data	zone_id	char [XD_MAX_STR]	Zone ID
		description	char [XD_MAX_STR]	Drescription of the zone
		surface	char [XD_MAX_STR]	Surface type
		creator	char [XD_MAX_STR]	Creator name
		zone_type	XD_Zone_type_enum	Zone type
		projection	long	Projection
		zone_diam	double	Zone diameter in meters. Only used when the ZONE is a POINT zone or a CIRCULAR zone.
		num_points	long	Number of ZONE points (last one, equal to the first one, included)
		zone_point	xd_zone_point *	Array of points of the zone
xd_zone_file	Zone file structure	num_rec	long	Number of zones
		zone_rec	xd_zone_rec *	Array of zone records
xd_scf_appear	Apearance data for swath configuration files	colour	long	Colour (hexadecimal value from 0x000000 to 0xFFFFFF)
		draw	long	Draw (see enumeration in Table 2)
		fill	long	Fill (see enumeration in Table 2)
		opacity	long	Opacity (0-100%)

Structure name	Description	Structure Data		
		Variable Name	C type	Description
xd_tle_rec	TLE record. It contains data for a TLE	norad_sat_cat	char[25]	Satellite name consistent with the NORAD SATCAT
		sat_number	long	NORAD Catalogue number
		classification	char	Classification: U=unclassified, S=secret data
		int_des	char [9]	International Designator: (Last two digits of launch year) (Launch number of the year) (Piece of the launch)
		time	double	reference time for the element set (UTC processing days MJ2000)
		n_1st	double	First Time Derivative of the Mean Motion
		n_2nd	double	Second Time Derivative of Mean Motion
		bstar	double	BSTAR drag term
		ephemeris_type	int	Ephemeris type
		index	int	Element number
		checksum1	int	Checksum for line 1
		i	double	inclination [Degrees]
		ra	double	Right Ascension of the Ascending Node [Degrees]
		e	double	Eccentricity
		w	double	Argument of Perigee [Degrees]
		m	double	Mean Anomaly [Degrees]
		n	double	Mean Motion [Revs per day]
		abs_orbit	long	Revolution number at epoch [Revs]
		checksum2	int	Checksum for line 2
xd_tle_file	Structure to store the data from a TLE file	num_rec	long	Number of records (TLE)
		tle_rec	xd_tle_rec*	Array with of TLE records
xd_propag_precise_config	Parameters for precise propagation configuration	user_flag	long	Indicates if default (0) or user defined (1) values are used for some parameters.
		models_path	char[256]	Path where files necessary for models are looked for.
		gravity_flag	long	Gravity perturbation used (1) or not (0).

Structure name	Description	Structure Data		
		Variable Name	**C type**	**Description**
		thirdbody_flag	long	Third bodies (Sun and Moon) perturbation used (1) or not (0).
		atmos_flag	long	Atmosphere perturbation used (1) or not (0).
		srp_flag	long	Solar radiation pressure perturbation used (1) or not (0).
		step	double	Simulation step (seconds).
		grav_file	char[256]	File with data of gravitational model.
		grav_degree	long	Degree used gravity model.
		grav_order	long	Order used in gravity model.
		sga_flag	long	ap, f107 and f107a parameters used (0) or data read from files sga_ap_file and sga_f107_file (1).
		sga_ap_file	char[256]	File with Geomagnetic Activity index values.
		sga_f107_file	char[256]	File with F10.7 Solar Activity index values.
		ap	double	Geomagnetic Activity Index (daily value).
		f107	double	F10.7 Index Solar Activity Index (daily value).
		f107a	double	F10.7 Index Solar Activity Index (value averaged over 3 months).
		sc_mass	double	S/C mass [kg].
		sc_drag_area	double	S/C effective drag area [m^2].
		sc_drag_coeff	double	S/C drag coefficient.
		sc_srp_area	double	S/C effective Solar Radiation Pressure area [m^2].
		sc_srp_coeff	double	S/C Solar Radiation Pressure coefficient.

7. CFI FUNCTIONS DESCRIPTION

The following sections describe each CFI function. The calling interfaces are described for C.

Input and output parameters of each CFI function are described in tables, where C programming language syntax is used to specify:

- Parameter types (e.g. long, double)
- Array sizes of N elements (e.g. param[N])
- Array element M (e.g. [M])

7.1.xd_read_fhr

7.1.1.Overview

The **xd_read_fhr** CFI function reads the fixed header for Earth ExplorerObservation XML files.

7.1.2.Calling interface

The calling interface of the **xd_read_fhr** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     char *file_name;
     xd_fhr fhr;
     long ierr[XD_NUM_ERR_READ_FHR];
     status = xd_read_fhr(file_name, &fhr, ierr);
}
```

7.1.3.Input parameters

The **xd_read_fhr** CFI function has the following input parameters:

Table 4: Input parameters of xd_read_fhr function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	file name	-	-

7.1.4.Output parameters

The output parameters of the **xd_read_orbit_file** CFI function are:

Table 5: Output parameters of xd_read_fhr function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_fhr	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
Fixed header data	xd_fhr	-	Data structure containing the data read from the fixed header	-	-
ierr	long[]	-	Error vector	-	-

7.1.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_fhr** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EOXPLORER_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_fhr** function by calling the function of the EOXPLORER_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 6: Error messages of xd_read_fhr function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the file	No calculation performed	XD_CFI_READ_FHR_OPEN_FILE_ERR	0
ERR	Error reading the fixed header	No calculation performed	XD_CFI_READ_FHR_GET_FIXED_HEADER_ERR	1
ERR	Error closing the file	No calculation performed	XD_CFI_READ_FHR_CLOSE_FILE_ERR	2

7.2.xd_read_bulletin

7.2.1.Overview

The **xd_read_bulletin** CFI function reads IERS bulletin files and returns the data relevant for time correlations. Either version 1980 as version 2010 of the IERS bulletins can be read.

This function is deprecated, it is recommended to use xd_read_bulletin_2.

7.2.2.Calling interface

The calling interface of the **xd_read_bulletin** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     char *bulb_file;
     xd_iers_bulletin_b iers_data
     long ierr[XD_NUM_ERR_READ_BULLETIN];
     status = xd_read_bulletin (bulb_file, &iers_data, ierr);
}
```

7.2.3.Input parameters

The **xd_read_bulletin** CFI function has the following input parameters:

Table 7: Input parameters of xd_read_bulletin function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
bulb_file	char*	-	File name	-	-

7.2.4.Output parameters

The output parameters of the **xd_read_bulletin** CFI function are:

Table 8: Output parameters of xd_read_bulletin function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_bulletin	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-

IERS bulletin data	xd_iers_bulletin_b	-	Data structure containing the data read from the file	-	-
ierr	long[]	-	Error vector	-	-

7.2.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_bulletin** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_bulletin** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 9: Error messages of xd_read_bulletin function

Error type	Error message	Cause and impact	Error code	Error No
ERR	File does not exist	No calculation performed	XD_CFI_READ_BULLETIN_FILE_ERR	0
ERR	Time table is empty or has wrong format	No calculation performed	XD_CFI_READ_BULLETIN_TABLE_ERR	1
ERR	File is not recognized	No calculation performed	XD_CFI_READ_BULLETIN_FILE_RECOG_ERR	2

7.3.xd_read_bulletin_2

7.3.1.Overview

The **xd_read_bulletin_2** CFI function reads IERS bulletin A and B files and returns the data relevant for time correlations and polar motion. Only version 2010 of the IERS bulletin B can be read.

7.3.2.Calling interface

The calling interface of the **xd_read_bulletin_2** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     char *bulb_file;
     xd_iers_bulletin_file iers_data
     long ierr[XD_NUM_ERR_READ_BULLETIN];
     status = xd_read_bulletin_2 (bulb_file, &iers_data, ierr);
}
```

7.3.3.Input parameters

The **xd_read_bulletin** CFI function has the following input parameters:

Table 10: Input parameters of xd_read_bulletin_2 function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
bulb_file	char*	-	File name	-	-

7.3.4.Output parameters

The output parameters of the **xd_read_bulletin** CFI function are:

Table 11: Output parameters of xd_read_bulletin_2 function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_bulletin_2	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-

IERS bulletin data	xd_iers_bulletin_file	-	Data structure containing the data read from the file	-	-
ierr	long[]	-	Error vector	-	-

7.3.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_bulletin_2** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_bulletin_2** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 12: Error messages of xd_read_bulletin_2 function

Error type	Error message	Cause and impact	Error code	Error No
ERR	File does not exist	No calculation performed	XD_CFI_READ_BULLETIN_FILE_ERR	0
ERR	Time table is empty or has wrong format	No calculation performed	XD_CFI_READ_BULLETIN_TABLE_ERR	1
ERR	File is not recognized	No calculation performed	XD_CFI_READ_BULLETIN_FILE_RECOG_ERR	2

7.4. xd_free_bulletin

7.4.1.Overview

The **xd_free_bulletin** CFI function frees the memory allocated during the reading function xd_read_bulletin_2.

7.4.2.Calling interface

The calling interface of the **xd_free_bulletin** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_iers_bulletin_file bulletin_data;
     xd_free_bulletin (&bulletin_data);
}
```

7.4.3.Input parameters

The **xd_free_bulletin** CFI function has the following input parameters:

Table 13: Input parameters of xd_free_bulletin function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
bulletin_data	xd_iers_bulletin_file	-	Bulletin file data structure	-	-

7.4.4.Output parameters

This function does not return any value nor parameters.

7.5.xd_read_orbit_file

7.5.1.Overview

The **xd_read_orbit_file** CFI function reads orbit files for Earth Observation Missions. The files have to be written in XML and consist on a list of state vectors of the satellite along the orbit.

This function can also be used for reading the list of state vectors within Orbit Event files.

7.5.2.Calling interface

The calling interface of the **xd_read_orbit_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     char *file_name;
     long read_fro_flag, time_orbit_flag, time_ref, reading_osv_flag;
     double start_range, stop_range;
     xd_orbit_file orbit_data
     long ierr[XD_NUM_ERR_READ_ORBIT_FILE];
     status = xd_read_orbit_file (file_name, &read_fro_flag,
                                  &time_orbit_flag, &time_ref,
                                  &start_range, &stop_range,
                                  &reading_osv_flag,
                                  &orbit_data, ierr);
}
```

7.5.3.Input parameters

The **xd_read_orbit_file** CFI function has the following input parameters:

Table 14: Input parameters of xd_read_orbit_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Orbit file name	-	-
read_fro_flag	long*	-	flag to indicate if the input file is: • a predicted orbit file • a restituted orbit file or a DORIS Preliminary file		• XD_TRUE for ROF and DORIS files • XD_FALSE for POF files

time_orbit_flag	long*	-	Flag for selecting the time range of the initialisation. Select either: • XD_SEL_FILE: for reading the whole file • XD_SEL_ORBIT: for reading the interval given by the start_range and the stop range parameters in orbits • XD_SEL_TIME: for reading the interval given by the start_range and the stop range parameters in days	-	All
time_ref	long*	-	Time reference if time_orbit_flag is XD_SEL_TIME. Dummy otherwise.	-	-
reading_osv_flag	long*	-	flag to indicate if the state vectors data have to be read.	-	• XD_TRUE for reading the state vector data • XD_FALSE for reading just the times and orbit numbers
start_range	double*	-	Start orbit or day	orbits or days	-
stop_range	double*	-	Stop orbit or day	orbits or days	-

It is possible to use enumeration values rather than integer values for some of the input arguments:

- Time model ID: time_model. See [GEN_SUM].
- Time reference ID: time_ref. See [GEN_SUM].
- Time range initialisation flag: time_orbit_flag. See current document, section 6.2

7.5.4.Output parameters

The output parameters of the **xd_read_orbit_file** CFI function are:

Table 15: Output parameters of xd_read_orbit_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_orbit_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
orbit_data	xd_orbit_file	-	Data structure containing the data read from the file	-	-

ierr	long[]	-	Error vector	-	-

Memory Management: The *orbit_data* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_orbit_file**

7.5.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_orbit_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_orbit_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 16: Error messages of xd_read_orbit_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error in reading file	No calculation performed	XD_CFI_READ_ORBIT_FILE_READ_ERR	0
ERR	Error in getting the first element inside the input range	No calculation performed	XD_CFI_READ_ORBIT_FILE_INPUT_RANGE_ERR	1
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_ORBIT_FILE_MEMORY_ERR	2
ERR	Internal Error # 1	No calculation performed	XD_CFI_READ_ORBIT_FILE_INTERNAL_1_ERR	3
ERR	Error while reading data	No calculation performed	XD_CFI_READ_ORBIT_FILE_DATA_READ_ERR	4
ERR	Gap found after OSV no. %li	No calculation performed	XD_CFI_READ_ORBIT_FILE_GAP_ERR	5

7.6.xd_free_orbit_file

7.6.1.Overview

The **xd_free_orbit_file** CFI function frees the memory allocated during the reading function xd_read_orbit_file.

7.6.2.Calling interface

The calling interface of the **xd_free_orbit_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_orbit_file orbit_data xd_free_orbit_file (&orbit_data);
}
```

7.6.3.Input parameters

The **xd_free_orbit_file** CFI function has the following input parameters:

Table 17: Input parameters of xd_free_orbit_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
orbit_data	xd_orbit_file	-	Orbit data structure	-	-

7.6.4.Output parameters

This function does not return any value nor parameters.

7.7.xd_read_doris

7.7.1.Overview

The **xd_read_doris** CFI function reads DORIS Navigator files for Cryosat.

7.7.2.Calling interface

The calling interface of the **xd_read_doris** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *doris_file;
     long time_mode, interpol_flag;
     double time0, time1;
     xd_doris_file doris_data
     long ierr[XD_NUM_ERR_READ_DORIS];

     status = xd_read_doris(doris_file, &time_mode,
                           &time0, &time1,
                           &interpol_flag,
                           &doris_data, ierr);
}
```

7.7.3.Input parameters

The **xd_read_doris** CFI function has the following input parameters:

Table 18: Input parameters of xd_read_doris function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
doris_file	char*	-	DORIS Navigator file name	-	-
time_mode	long	-	Flag for reading the whole file or just the requested time window	-	• XD_SEL_FILE • XD_SEL_TIME
time0	double	-	Start time for the requested time window (if XD_SEL_TIME selected)	days in UTC	-
time1	double	-	Stop time for the requested time window (if XD_SEL_TIME selected)	days in UTC	-

interpol_flag	long	-	Flag to indicate if the read data are used for interpolation purposes. In that case 4 extra state vectors are read out of the requested time window	-	• XD_TRUE for interpol data • XD_FALSE otherwise

It is possible to use enumeration values rather than integer values for some of the input arguments:

- Time model ID: time_mode. See [GEN_SUM].

7.7.4.Output parameters

The output parameters of the **xd_read_doris** CFI function are:

Table 19: Output parameters of xd_read_doris function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_doris	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
doris_data	xd_doris_file	-	DORIS data	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *doris_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_doris**.

7.7.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_doris** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_doris** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 20: Error messages of xd_read_doris function

Error type	Error message	Cause and impact	Error code	Error No

ERR	Error in memory assignation	No calculation performed	XD_CFI_READ_DORIS_ERROR_IN_MEMORY_ASIG_ERR	0
ERR	Wrong input parameter value: \"time_mode\"	No calculation performed	XD_CFI_READ_DORIS_WRONG_TIME_MODE_ERR	1
ERR	Wrong time on input (start time after stop time)	No calculation performed	XD_CFI_READ_DORIS_WRONG_TIME_1_ERR	2
ERR	Wrong time on input (out of limits)	No calculation performed	XD_RCFI_EAD_DORIS_WRONG_TIME_2_ERR	3
ERR	DORIS level 0 filename not supplied	No calculation performed	XD_CFI_READ_DORIS_NO_FILENAME_ERR	4
ERR	DORIS Level 0 file cannot be open	No calculation performed	XD_CFI_READ_DORIS_CANNOT_OPEN_ERR	5
ERR	Could not find keyword: %s	No calculation performed	XD_CFI_READ_DORIS_FINDKW_ERROR_ERR	6
ERR	Error reading DORIS data for keyword: %s	No calculation performed	XD_CFI_READ_DORIS_READ_ERR	7
ERR	Error reading DORIS binary data	No calculation performed	XD_CFI_READ_DORIS_READ_BIN_ERR	8
ERR	Error changing time from ascii to processing	No calculation performed	XD_CFI_READ_DORIS_ASCII_TO_PROCESSING_ERR	9
ERR	Gap found reading DORIS level0 data	No calculation performed	XD_CFI_READ_DORIS_GAP_IN_FILE_ERR	10
ERR	DORIS file does not cover user required time interval	No calculation performed	XD_CFI_READ_DORIS_DOES_NOT_COVER_TIME_INTERVAL_ERR	11

7.8.xd_free_doris

7.8.1.Overview

The **xd_free_doris** CFI function frees the memory allocated during the reading function **xd_read_doris**.

7.8.2.Calling interface

The calling interface of the **xd_free_doris** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_doris_file doris_data xd_free_doris (&doris_data);
}
```

7.8.3.Input parameters

The **xd_free_doris** CFI function has the following input parameters:

Table 21: Input parameters of xd_free_doris function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
doris_data	xd_doris_file	-	DORIS data structure	-	-

7.8.4.Output parameters

This function does not return any value nor parameters.

7.9.xd_read_doris_header

7.9.1.Overview

The **xd_read_doris_header** CFI function reads the Main Product Header (MPH) and the Specific Product Header (SPH) from DORIS Navigator files for Cryosat.

7.9.2.Calling interface

The calling interface of the **xd_read_doris_header** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *doris_file;
     xd_doris_mph_sph doris_hdr;
     long ierr[XD_NUM_ERR_READ_DORIS_HEADER];

     status = xd_read_doris_header(doris_file, &doris_hdr, ierr);
}
```

7.9.3.Input parameters

The **xd_read_doris_header** CFI function has the following input parameters:

Table 22: Input parameters of xd_read_doris_header function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
doris_file	char*	-	DORIS file name	-	-

It is possible to use enumeration values rather than integer values for some of the input arguments:

- Time model ID: time_mode. See [GEN_SUM].

7.9.4.Output parameters

The output parameters of the **xd_read_doris_header** CFI function are:

Table 23: Output parameters of xd_read_doris_header function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_doris_header	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
doris_data	xd_doris_mph_sph	-	doris header structure	-	-
ierr	long []	.	Error vector	-	-

7.9.5. Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_doris_header** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_doris_header** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 24: Error messages of xd_read_doris function

Error type	Error message	Cause and impact	Error code	Error No
ERR	DORIS level 0 filename not supplied	No calculation performed	XD_CFI_READ_DORIS_HEADER_NO_FILENAME_ERR	0
ERR	DORIS Level 0 file cannot be open	No calculation performed	XD_CFI_READ_DORIS_HEADER_CANNOT_OPEN_ERR	1
ERR	Could not find keyword: %s	No calculation performed	XD_CFI_READ_DORIS_HEADER_FINDKW_ERROR_ERR	2
ERR	Error reading DORIS data for keyword: %s	No calculation performed	XD_CFI_READ_DORIS_HEADER_READ_ERR	3

7.10. xd_read_osf

7.10.1. Overview

The **xd_read_osf** CFI function reads Orbit Scenario files for Earth Observation Missions. The files have to be written in XML and consist on a list of orbital changes of the satellite along the orbit.

This function can also be used for reading the list of orbital changes within Orbit Event files.

7.10.2. Calling interface

The calling interface of the **xd_read_osf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_osf_file osf_data;
     long ierr[XD_NUM_ERR_READ_OSF];

     status = xd_read_osf (file_name, &osf_data, ierr);
}
```

7.10.3. Input parameters

The **xd_read_osf** CFI function has the following input parameters:

Table 25: Input parameters of xd_read_osf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Orbit Scenario file name	-	-

7.10.4. Output parameters

The output parameters of the **xd_read_osf** CFI function are:

Table 26: Output parameters of xd_read_osf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range

xd_read_osf	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
osf_data	xd_osf_file	-	Structure with the OSF data	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *osf_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_osf**.

7.10.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_osf** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_osf** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 27: Error messages of xd_read_osf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error initializing the file parser	No calculation performed	XD_CFI_READ_XML_OSF_INIT_PARSER_ERR	0
ERR	Error finding the data block keyword	No calculation performed	XD_CFI_READ_XML_OSF_XML_DATA_BLOCK_ERR	1
ERR	Error reading the data block attribute	No calculation performed	XD_CFI_READ_XML_OSF_XML_ATTRIBUTE_ERR	2
ERR	"Error reading the xml attribute"	No calculation performed	XD_CFI_READ_XML_OSF_XML_TYPE_ERR	3
ERR	Error reading XML element: %s	No calculation performed	XD_CFI_READ_XML_OSF_READ_PARAM_ERR	4
ERR	Error the size of the list (negative)	No calculation performed	XD_CFI_READ_XML_OSF_XML_DATA_BLOCK_SIZE_ERR	5

ERR	Error allocating memory	No calculation performed	XD_CFI_READ_XML_OSF_MEMORY_ERR	6
ERR	Variable header not found	No calculation performed	XD_CFI_READ_XML_OSF_VHR_NOT_FOUND_ERR	7
ERR	Incorrect value of Time_Reference. OSF time reference must be UT1	No calculation performed	XD_CFI_READ_XML_OSF_TIME_REF_OF_ERR	8
WARN	No time reference specified in orbit scenario file. Assuming UT1	Calculation performed	XD_CFI_READ_XML_OSF_TIME_REF_OF_WARN	9

7.11. xd_free_osf

7.11.1.Overview

The **xd_free_osf** CFI function frees the memory allocated during the reading function **xd_read_osf**.

7.11.2.Calling interface

The calling interface of the **xd_free_osf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_osf_file osf_data xd_free_osf (&osf_data);
}
```

7.11.3.Input parameters

The **xd_free_osf** CFI function has the following input parameters:

Table 28: Input parameters of xd_free_osf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
osf_data	xd_osf_file	-	DORIS data structure	-	-

7.11.4.Output parameters

This function does not return any value nor parameters.

7.12. xd_read_sdf

7.12.1.Overview

The **xd_read_sdf** CFI function reads Swath Definition files for Earth Observation Missions. For compatibility, it is possible to read files with old format.

7.12.2.Calling interface

The calling interface of the **xd_read_sdf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status; xd_sdf_file sdf_data; char *file_name;
     long ierr[XD_NUM_ERR_READ_SDF];

     status = xd_read_sdf (file_name, &sdf_data, ierr);
}
```

7.12.3.Input parameters

The **xd_read_sdf** CFI function has the following input parameters:

Table 29: Input parameters of xd_read_sdf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Swath Definition file name	-	-

7.12.4.Output parameters

The output parameters of the **xd_read_sdf** CFI function are:

Table 30: Output parameters of xd_read_sdf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_sdf	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
sdf_data	xd_sdf_file	-	Swath Definition data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *sdf_data* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_sdf**.

7.12.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_sdf** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_sdf** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 31: Error messages of xd_read_sdf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error opening Swath Definition file	No calculation performed	XD_CFI_READ_SDF_OPEN_FILE_ERR	0
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_SDF_MEMORY_ERR	1
ERR	Error reading swath record %d	No calculation performed	XD_CFI_READ_SDF_REC_READ_ERR	2
ERR	Could not get file version	No calculation performed	XD_CFI_READ_SDF_VERSION_ERR	3

7.13.xd_free_sdf

7.13.1.Overview

The **xd_free_sdf** CFI function frees the memory allocated during the reading function **xd_read_sdf**.

7.13.2.Calling interface

The calling interface of the **xd_free_sdf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_sdf_file sdf_data xd_free_sdf (&sdf_data);
}
```

7.13.3.Input parameters

The **xd_free_sdf** CFI function has the following input parameters:

Table 32: Input parameters of xd_free_sdf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sdf_data	xd_sdf_file	-	SDF data structure	-	-

7.13.4.Output parameters

This function does not return any value nor parameters.

7.14. xd_read_stf

7.14.1.Overview

The **xd_read_stf** CFI function reads Swath Template Files for Earth Observation Missions. For compatibility, it is possible to read files with old format.

7.14.2.Calling interface

The calling interface of the **xd_read_stf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_stf_file stf_data;
     long ierr[XD_NUM_ERR_READ_STF];

     status = xd_read_stf (file_name, &stf_data, ierr);
}
```

7.14.3.Input parameters

The **xd_read_stf** CFI function has the following input parameters:

Table 33: Input parameters of xd_read_stf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Swath Template file name	-	-

7.14.4.Output parameters

The output parameters of the **xd_read_stf** CFI function are:

Table 34: Output parameters of xd_read_stf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_stf	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
stf_data	xd_stf_file	-	Swath template file data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *stf_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_stf**.

7.14.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_stf** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_stf** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 35: Error messages of xd_read_stf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error initializing parser to read the file	No calculation performed	XD_CFI_READ_STF_INIT_PARSER_ERR	0
ERR	Error reading the variable header	No calculation performed	XD_READ_STF_VHR_ERR	1
ERR	Error reading element: %s"	No calculation performed	XD_CFI_READ_STF_PARAM_READ_ERR	2
ERR	Could not find data block.	No calculation performed	XD_CFI_READ_STF_DATA_BLOCK_ERR	3
ERR	Could not read Data_Block attribute.	No calculation performed	XD_CFI_READ_STF_ATTRIBUTE_ERR	4

ERR	Data block is not XML type.	No calculation performed	XD_CFI_READ_STF_XML_TYPE_ERR	5
ERR	Negative number of swath coordinates	No calculation performed	XD_CFI_READ_STF_DATA_BLOCK_SIZE_ERR	6
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_STF_MEMORY_ERR	7
ERR	Error reading swath record #%d	No calculation performed	XD_CFI_READ_STF_REC_READ_ERR	8
ERR	Error in STF, latitude/Dec out of range for swath record #%ld	No calculation performed	XD_CFI_READ_STF_WRONG_LAT_ERR	9
ERR	Error in STF, longitude/RA out of range for swath record #%ld	No calculation performed	XD_CFI_READ_STF_WRONG_LONG_ERR	10

7.15.xd_free_stf

7.15.1.Overview

The **xd_free_stf** CFI function frees the memory allocated during the reading function **xd_read_stf**.

7.15.2.Calling interface

The calling interface of the **xd_free_stf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_stf_file stf_data xd_free_stf (&stf_data);
}
```

7.15.3.Input parameters

The **xd_free_stf** CFI function has the following input parameters:

Table 36: Input parameters of xd_free_stf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
stf_data	xd_stf_file	-	STF data structure	-	-

7.15.4.Output parameters

This function does not return any value nor parameters.

7.16.xd_read_stf_vhr

7.16.1.Overview

The **xd_read_stf_vhr** CFI function reads the variable header in Swath Template File for Earth Observation Missions.

7.16.2.Calling interface

The calling interface of the **xd_read_stf_vhr** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_stf_vhr vhr_data;
     long ierr[XD_NUM_ERR_READ_STF_VHR];

     status = xd_read_stf_vhr (file_name, &vhr_data, ierr);
}
```

7.16.3.Input parameters

The **xd_read_stf_vhr** CFI function has the following input parameters:

Table 37: Input parameters of xd_read_stf_vhr function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*^	-	Swath Template file name	-	-

7.16.4.Output parameters

The output parameters of the **xd_read_stf_vhr** CFI function are:

Table 38: Output parameters of xd_read_stf_vhr function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_stf_vhr	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
vhr_data	xd_stf_vhr	-	Data structure for the Swath template variable header	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *vhr_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_stf_vhr**.

7.16.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_stf_vhr** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_stf_vhr** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 39: Error messages of xd_read_stf_vhr function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error initializing parser to read the file	No calculation performed	XD_CFI_READ_STF_VHR_INIT_PARSER_ERR	0
ERR	Could not find variable header	No calculation performed	XD_CFI_READ_STF_VHR_VARIABLE_HEADER_ERR	1
ERR	Error within the reading function	No calculation performed	XD_CFI_READ_STF_VHR_INTERNAL_1_ERR	2
ERR	Error reading element: %s	No calculation performed	XD_CFI_READ_STF_VHR_PARAM_READ_ERR	3
ERR	Incorrect swath type	No calculation performed	XD_CFI_READ_STF_VHR_SWATH_TYPE_ERR	4

ERR	Incorrect swath point type	No calculation performed	XD_CFI_READ_STF_VHR_SWATH_POINT_TYPE_ERR	5
ERR	Error reading "Orbit_State_Vector"	No calculation performed	XD_CFI_READ_STF_VHR_ORBIT_PARAMS_ERR	6
ERR	Error reading "Orbit_Geometry”	No calculation performed	XD_CFI_READ_STF_VHR_GEOM_PARAMS_ERR	7
ERR	Error reading altitude	No calculation performed	XD_CFI_READ_STF_VHR_ALTITUDE_READ_ERR	8
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_STF_VHR_MEMORY_ERR	9

7.17. xd_free_stf_vhr

7.17.1.Overview

The **xd_free_stf_vhr** CFI function frees the memory allocated during the reading function xd_read_stf_vhr.

7.17.2.Calling interface

The calling interface of the **xd_free_stf_vhr** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_stf_vhr stf_vhr;
     xd_free_stf_vhr (&stf_vhr);
}
```

7.17.3.Input parameters

The **xd_free_stf_vhr** CFI function has the following input parameters:

Table 40: Input parameters of xd_free_stf_vhr function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
stf_vhr	xd_stf_vhr	-	STF variable header data structure	-	-

7.17.4.Output parameters

This function does not return any value nor parameters.

7.18.xd_read_att

7.18.1.Overview

The xd_read_att CFI function reads attitude generic files. This files have to be written in XML and consits on a list of attitude angles or quaternions.

7.18.2.Calling interface

The calling interface of the **xd_read_att** CFI function is the following (input parameters are <u>underlined</u>):

```
#include <explorer_data_handling.h>
{
     long status; xd_att_file att_data; char *file_name;
     long ierr[XD_NUM_ERR_READ_ATT];

     status = xd_read_att (file_name, att_data, ierr);
}
```

7.18.3.Input parameters

The **xd_read_att** CFI function has the following input parameters:

Table 41: Input parameters of xd_read_att function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Attitude file name	-	-

7.18.4.Output parameters

The output parameters of the **xd_read_** CFI function are:

Table 42: Output parameters of xd_read_att function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowd Range

xd_read_att	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
att_data	xd_att_file	-	Attitude data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *att_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_att**.

7.18.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_att** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_att** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 43: Error messages of xd_read_att function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error initializing parser to read the file	No calculation performed	XD_CFI_READ_ATT_INIT_PARSER_ERR	0
ERR	Error reading element: %s	No calculation performed	XD_CFI_READ_ATT_READ_PARAM_ERR	1
ERR	Wrong file type	No calculation performed	XD_CFI_READ_ATT_WRONG_FILE_TYPE_ERR	2
ERR	Error navigating through the file	No calculation performed	XD_CFI_READ_XML_ATT_NAVIGATION_ERR	3
ERR	Wrong attitude data type. Only "Quaternions" and "Attitude_Angles_Data" allowed	No calculation performed	XD_CFI_READ_ATT_WRONG_DATA_TYPE_ERR	4

ERR	Inconsistent values for <Attitude_Data_Type> and the list of attitude data	No calculation performed	XD_CFI_READ_ATT_INCONSISTENT_DATA_TYPE_ERR	5
ERR	Wrong number of records in the list	No calculation performed	XD_CFI_READ_ATT_XML_DATA_BLOCK_SIZE_ERR	6
ERR	Wrong parameter in "Inertial_Ref_Frame"	No calculation performed	XD_CFI_READ_ATT_WRONG_REF_FRAME_ERR	7
ERR	Error reading attitude data list	No calculation performed	XD_CFI_READ_ATT_READ_LIST_ERR	8
ERR	Error converting ascii date to processing	No calculation performed	XD_CFI_READ_ATT_TIME_CONV_ERR	9
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_ATT_MEMORY_ERR	10
ERR	Could not close the file	No calculation performed	XD_CFI_READ_ATT_CLEANUP_PARSER_ERR	11
ERR	Wrong time reference for element n. %d. All time references should be equal	No calculation performed	XD_CFI_READ_ATT_WRONG_TIME_REF_ERR	12
ERR	Quaternion modulus out of limits. Check list element n.%d	No calculation performed	XD_CFI_READ_ATT_WRONG_QUATERNION_ERR	13
ERR	Angle out of limits. Check list element n. %d	No calculation performed	XD_CFI_READ_ATT_WRONG_ANGLE_ERR	14
ERR	Maximum Gap value must be positive	No calculation performed	XD_CFI_READ_ATT_MAX_GAP_ERR	15

7.19.xd_free_att

7.19.1.Overview

The **xd_free_att** CFI function frees the memory allocated during the reading function **xd_read_att**.

7.19.2.Calling interface

The calling interface of the **xd_free_att** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
      xd_att_file att_data;
      xd_free_att (&att_data);
}
```

7.19.3.Input parameters

The **xd_free_att** CFI function has the following input parameters:

Table 44: Input parameters of xd_free_att function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
att_data	xd_att_file	-	Attitude data structure	-	-

7.19.4.Output parameters

This function does not return any value nor parameters.

7.20.xd_read_star_tracker

7.20.1.Overview

The **xd_read_star_tracker** CFI function reads a list of star tracker files for Cryosat.

7.20.2.Calling interface

The calling interface of the **xd_read_star_tracker** CFI function is the following (input parameters are <u>underlined</u>):

```
#include <explorer_data_handling.h>
{
     long status;
     long n_files, time_init_mode;
     char **file_list;
     double time0, time1;
     xd_tracker_limits str_limit;
     xd_star_tracker_file str_data;
     long ierr[XD_NUM_ERR_READ_STAR_TRACKER];

     status = xd_read_star_tracker (&n_files, file_list,
                              &time_init_mode, &time0, &time1,
                              &str_limit,
                              &str_data, ierr);
}
```

7.20.3.Input parameters

The **xd_read_star_tracker** CFI function has the following input parameters:

Table 45: Input parameters of xd_read_star_tracker function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
n_files	long	-	Number of input files	-	> 0
file_list	char **	-	List of star tracker files	-	-
time_init_mode	long	-	Flag for reading the whole file or just the requested time window		• XD_SEL_FILE or • XD_SEL_TIME
time0	double	-	Start time for the requested time window	-	days (TAI)

time1	double	-	Stop time for the requested time window	-	days (TAI)
str_limit	xd_str_limits	-	data structure containing the limits for the quaternion validation	-	-

It is possible to use enumeration values rather than integer values for some of the input arguments:

- Time range initialisation flag: time_init_mode. See current document, section 6.2

7.20.4.Output parameters

The output parameters of the **xd_read_star_tracker** CFI function are:

Table 46: Output parameters of xd_read_star_tracker function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_star_tracker	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
str_data	xd_star_tracker_file	-	Star tracker data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *str_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_star_tracker**.

7.20.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_star_tracker** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_star_tracker** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 47: Error messages of xd_read_star_tracker function

Error type	**Error message**	**Cause and impact**	**Error code**	**Error No**
ERR	Could not open input file	No calculation performed	XD_CFI_READ_STR_TRACKER_OPEN_FILE_ERR	0
ERR	Could not read input file	No calculation performed	XD_CFI_READ_STR_TRACKER_READ_FILE_ERR	1
ERR	Memory allocation error	No calculation performed	XD_CFI_READ_STR_TRACKER_MEMORY_FILE_ERR	2
ERR	Gap between quaternions above maximum allowed value after time %f	No calculation performed	XD_CFI_READ_STR_TRACKER_GAP_ERR	3
ERR	No enough valid quaternions to cover the requested interval	No calculation performed	XD_CFI_READ_STR_TRACKER_NO_ENOUGH_DATA_ERR	4

7.21.xd_free_star_tracker

7.21.1.Overview

The **xd_free_star_tracker** CFI function frees the memory allocated during the reading function xd_read_star_tracker.

7.21.2.Calling interface

The calling interface of the **xd_free_star_tracker** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_star_tracker_file str_data;
     xd_free_star_tracker (&str_data);
}
```

7.21.3.Input parameters

The **xd_free_star_tracker** CFI function has the following input parameters:

Table 48: Input parameters of xd_free_star_tracker function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
str_data	xd_star_tracker_file	-	Star tracker data structure	-	-

7.21.4.Output parameters

This function does not return any value nor parameters.

7.22.xd_read_star_tracker_conf_file

7.22.1.Overview

The **xd_read_star_tracker_conf_file** CFI function reads an star tracker configuration file for Cryosat. The files have to be written in XML.

7.22.2.Calling interface

The calling interface of the **xd_read_star_tracker_conf_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status, star_tracker_id;
     char *file_name;
     xd_tracker_conf_file conf_data;
     long ierr[XD_NUM_ERR_READ_STAR_TRACKER_CONF_FILE];

     status = xd_read_star_tracker_conf_file (file_name,
                                              &star_tracker_id,
                                              &conf_data, ierr);
}
```

7.22.3.Input parameters

The **xd_read_star_tracker_conf_file** CFI function has the following input parameters:

Table 49: Input parameters of xd_read_star_tracker_conf_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Star Tracker configuration file name	-	-
star_tracker_id	long	-	Star tracker number for which the configuration data is to be read	-	1, 2 or 3

7.22.4.Output parameters

The output parameters of the **xd_read_star_tracker_conf_file** CFI function are:

Table 50: Output parameters of xd_read_star_tracker_conf_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_star_tracker_conf_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
conf_data	xd_tracker_conf_file	-	Star tracker configuration data structure with	-	-
ierr	long[]	-	Error vector	-	-

7.22.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_star_tracker_conf_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_star_tracker_conf_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 51: Error messages of xd_read_star_tracker_conf_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input file	No calculation performed	XD_CFI_READ_STR_CONF_FILE_READ_FILE_ERR	0

7.23. xd_read_dem

7.23.1.Overview

The **xd_read_dem** CFI function reads a DEM file providing the table with the altitudes for each point of the grid of the DEM file.

7.23.2.Calling interface

The calling interface of the **xd_read_dem** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *dem_name;
     xd_dem_config_file dem_conf_data;
     xd_dem_file dem_data;
     long ierr[XD_NUM_ERR_READ_DEM];

     status = xd_read_dem (dem_name, &dem_conf_data,
                           &dem_data, ierr);
}
```

7.23.3.Input parameters

The **xd_read_dem** CFI function has the following input parameters:

Table 52: Input parameters of xd_read_dem function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
dem_name	char*	-	DEM file name (do not include the path)	-	-
dem_conf_data	xd_dem_config_file	-	DEM configuration data structure. This data are read from a configuration file with **xd_read_dem_config_file**	-	-

It is possible to use enumeration values rather than integer values for some of the input arguments:

- Time model ID: time_model. See [GEN_SUM].
- Time reference ID: time_ref. See [GEN_SUM].
- Time range initialisation flag: time_init_mode. See current document, section 6.2

7.23.4.Output parameters

The output parameters of the **xd_read_dem** CFI function are:

Table 53: Output parameters of xd_read_dem function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_dem	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
dem_data	xd_dem_file	-	DEM data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *dem_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_dem**.

7.23.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_dem** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_dem** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 54: Error messages of xd_read_dem function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Memory allocation error	No calculation performed	XD_CFI_READ_DEM_MEMORY_ERR	0
ERR	Incorrect input DEM configuration file	No calculation performed	XD_CFI_READ_DEM_NO_CONFIG_FILE_ERR	1
ERR	Wrong input file name	No calculation performed	XD_CFI_READ_DEM_WRONG_FILENAME_ERR	2
ERR	Could not open the DEM file	No calculation performed	XD_CFI_READ_DEM_OPEN_FILE_ERR	3

ERR	Could not read the DEM file	No calculation performed	XD_CFI_READ_DEM_READ_FILE_ERR	4
ERR	Unknown DEM model	No calculation performed	XD_READ_DEM_UNKNOWN_MODEL_ERR	5

7.24.xd_free_dem

7.24.1.Overview

The **xd_free_dem** CFI function frees the memory allocated in the reading function **xd_read_dem**.

7.24.2.Calling interface

The calling interface of the **xd_free_dem** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_dem_file dem_data;
     xd_free_dem (&dem_data);
}
```

7.24.3.Input parameters

The **xd_free_dem** CFI function has the following input parameters:

Table 55: Input parameters of xd_free_dem function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
dem_data	xd_dem_file	-	DEM data structure	-	-

7.24.4.Output parameters

This function does not return any value nor parameters.

7.25.xd_read_dem_config_file

7.25.1.Overview

The **xd_read_dem_config_file** CFI function reads DEM configuration parameters. These parameters are described in section 9.12. Note that the DEM version (1 or 2) is automatically detected (See [MCD] for further details about the DEM models).

7.25.2.Calling interface

The calling interface of the **xd_read_dem_config_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_dem_config_file dem_config_data;
     long ierr[XD_NUM_ERR_READ_DEM_CONFIG];

     status = xd_read_dem_config_file (file_name,
                                       &dem_config_data,
                                       ierr);
}
```

7.25.3.Input parameters

The **xd_read_dem_config_file** CFI function has the following input parameters:

Table 56: Input parameters of xd_read_dem_config_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	DEM configuration file name	-	-

7.25.4.Output parameters

The output parameters of the **xd_read_dem_config_file** CFI function are:

Table 57: Output parameters of xd_read_dem_config_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_dem_config_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
dem_config_data	xd_dem_config_file	-	DEM configuration data structure	-	-
ierr	long[]	-	Error vector	-	-

7.25.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_dem_config_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_dem_config_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 58: Error messages of xd_read_dem_config_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the configura tion file	No calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_OPEN_ERR	0
ERR	Could not read the configura tion file	No calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_READ_ERR	1
ERR	Could not open the model tag	No calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_READ_MODEL_ERR	2
ERR	Memory allocation error	No calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_MEMORY_ERR	3
WARN	Could not open a ACE Pole file	Calculation performed. Default value is taken.	XD_CFI_READ_DEM_CONFIG_FILE_OPEN_DEM_FILE_WARN	4
ERR	Could not read a ACE file	No calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_READ_DEM_FILE_ERR	5

WARN	Inpput DEM configuration file version is deprecated	Calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_DEPRECATED_WARN	6
WARN	DEM Cache Type not supplied, assuming FIFO_CACHE with maximum size of 2 GB	Calculation performed	XD_CFI_READ_DEM_CONFIG_FILE_CACHE_TYPE_WARN	7

7.26. xd_read_zone

7.26.1.Overview

The **xd_read_zone** CFI function reads a specific zone from a zone database file for Earth Observation Missions.

7.26.2.Calling interface

The calling interface of the **xd_read_zone** CFI function is the following (input parameters are underlined)

```
#include <explorer_data_handling.h>
{
     long status; char *zone_id; char *file_name;
     xd_zone_rec zone_rec;
     long ierr[XD_NUM_ERR_READ_ZONE];

     status = xd_read_zone (file_name, &zone_id, &zone_rec, ierr);
}
```

7.26.3.Input parameters

The **xd_read_zone** CFI function has the following input parameters:

Table 59: Input parameters of xd_read_zone function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Zone database file name	-	-
zone_id	char*	-	Zone Id to be read	-	-

7.26.4.Output parameters

The output parameters of the **xd_read_zone** CFI function are:

Table 60: Output parameters of xd_read_zone function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_zone	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
zone_rec	xd_zone_rec	-	Zone Data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *zone_rec* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_zone**.

7.26.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_zone** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_zone** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 61: Error messages of xd_read_zone function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Zone File not found	No calculation performed	XD_CFI_READ_ZONE_INIT_PARSER_ERR	0
ERR	Data Block not found	No calculation performed	XD_CFI_READ_ZONE_DATA_BLOCK_ERR	1
ERR	Data Block attribute not read	No calculation performed	XD_CFI_READ_ZONE_DATA_BLOCK_ATTRIBUTE_ERR	2
ERR	Data Block not of XML type	No calculation performed	XD_CFI_READ_ZONE_XML_TYPE_ERR	3

ERR	List_of_Zones not found.	No calculation performed	XD_CFI_READ_ZONE_LIST_ZONES_READ_ERR	4
ERR	List_of_Zones attribute not read.	No calculation performed	XD_CFI_READ_ZONE_LIST_ZONES_SIZE_ERR	5
ERR	Internal error returned	No calculation performed	XD_CFI_READ_ZONE_INTERNAL_1_ERR	6
ERR	Zone_ID cannot be read.	No calculation performed	XD_CFI_READ_ZONE_ZONE_ID_READ_ERR	7
ERR	Zone_ID not found.	No calculation performed	XD_CFI_READ_ZONE_ZONE_ID_NOT_FOUND_ERR	8
ERR	Error reading zone record	No calculation performed	XD_CFI_READ_ZONE_RECORD_READ_ERR	9

7.27.xd_free_zone

7.27.1.Overview

The **xd_free_zone** CFI function frees the memory allocated during the reading function **xd_read_zone**.

7.27.2.Calling interface

The calling interface of the **xd_free_zone** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
      xd_zone_rec zone_data;
      xd_free_zone (&zone_data);
}
```

7.27.3.Input parameters

The **xd_free_zone** CFI function has the following input parameters:

Table 62: Input parameters of xd_free_zone function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
zone_data	xd_zone_rec	-	Zone record data structure	-	-

7.27.4.Output parameters

This function does not return any value nor parameters.

7.28. xd_read_zone_file

7.28.1.Overview

The **xd_read_zone_file** CFI function reads a zone database file for Earth Observation Missions.

7.28.2.Calling interface

The calling interface of the **xd_read_zone_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_zone_file zone_data;
     long ierr[XD_NUM_ERR_READ_ZONE_FILE];

     status = xd_read_zone_file (file_name, &zone_data, ierr);
}
```

7.28.3.Input parameters

The **xd_read_zone_file** CFI function has the following input parameters:

Table 63: Input parameters of xd_read_zone_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Zone database file name	-	-

7.28.4.Output parameters

The output parameters of the **xd_read_zone_file** CFI function are:

Table 64: Output parameters of xd_read_zone_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_zone_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
xd_zone_file	zone_data	-	Structure containing the data for all the zones read from the file	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *zone_data* structure contains pointers to memory allocated dynamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_zone_file**.

7.28.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_zone_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_zone_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 65: Error messages of xd_read_zone_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Zone File not found.	No calculation performed	XD_CFI_READ_ZONE_FILE_INIT_PARSER_ERR	0
ERR	Data Block not found	No calculation performed	XD_CFI_READ_ZONE_FILE_DATA_BLOCK_ERR	1
ERR	Data Block attribute not read.	No calculation performed	XD_CFI_READ_ZONE_FILE_DATA_BLOCK_ATTRIBUTE_ERR	2
ERR	Data Block not of XML type.	No calculation performed	XD_CFI_READ_ZONE_FILE_XML_TYPE_ERR	3

ERR	List_of_Zones not found.	No calculation performed	XD_CFI_READ_ZONE_FILE_LIST_ZONES_READ_ERR	4
ERR	List_of_Zones attribute not read	No calculation performed	XD_CFI_READ_ZONE_FILE_LIST_ZONES_SIZE_ER R	5
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_ZONE_FILE_MEM_ERR	6
ERR	Error reading zone record number %d	No calculation performed	XD_CFI_READ_ZONE_FILE_RECORD_READ_ERR	7

7.29. xd_free_zone_file

7.29.1.Overview

The **xd_free_zone_file** CFI function frees the memory allocated during the reading function xd_read_zone_file.

7.29.2.Calling interface

The calling interface of the **xd_free_zone_file** CFI function is the following (input parameters are <u>underlined</u>):

```
#include <explorer_data_handling.h>
{
     xd_zone_file zone_data;
     xd_free_zone_file (&zone_data);
}
```

7.29.3.Input parameters

The **xd_free_zone_file** CFI function has the following input parameters:

Table 66: Input parameters of xd_free_zone_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
zone_data	xd_zone_file	-	Zone file data structure	-	-

7.29.4.Output parameters

This function does not return any value nor parameters.

7.30.xd_read_zone_id

7.30.1.Overview

The **xd_read_zone_id** CFI function reads the list of zone names (Id) in a zone database file for Earth Observation Missions.

7.30.2.Calling interface

The calling interface of the **xd_read_zone_id** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status, num_zones;
     char *file_name;
     char **zone_ids
     long ierr[XD_NUM_ERR_READ_ZONE_ID];

     status = xd_read_zone_id (file_name,
                               &num_zones, &zoned_ids,
                               ierr);
}
```

7.30.3.Input parameters

The **xd_read_zone_id** CFI function has the following input parameters:

Table 67: Input parameters of xd_read_zone_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*^	-	Zone database file name	-	-

7.30.4.Output parameters

The output parameters of the **xd_read_zone_id** CFI function are:

Table 68: Output parameters of xd_read_zone_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range

xd_read_zone_id	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
num_zones	long	-	Number of zones in the input file	-	-
zone_ids	char**	-	List fo zone names in the file	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *zone_ids* is a double pointer to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_zone_id**.

7.30.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_zone_id** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_zone_id** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 69: Error messages of xd_read_zone_id function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Zone File not found.	No calculation performed	XD_CFI_READ_ZONE_ID_INIT_PARSER_ERR	0
ERR	Data Block not found	No calculation performed	XD_CFI_READ_ZONE_ID_DATA_BLOCK_ERR	1
ERR	List_of_Zones not found.	No calculation performed	XD_CFI_READ_ZONE_ID_LIST_ZONES_READ_ERR	2
ERR	List_of_Zones attribute not read.	No calculation performed	XD_CFI_READ_ZONE_ID_LIST_ZONES_SIZE_ERR	3
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_ZONEI_D_MEMORY_ERR	4
ERR	Could not find the Zone_Id tag	No calculation performed	XD_CFI_READ_ZONE_ID_READ_ZONE_ERR	5

7.31. xd_free_zone_id

7.31.1.Overview

The **xd_free_zone_id** CFI function frees the memory allocated during the reading function xd_read_zone_id.

7.31.2.Calling interface

The calling interface of the **xd_free_zone_id** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     char** zone_ids;
     xd_free_zone_id (&zone_ids);
}
```

7.31.3.Input parameters

The **xd_free_zone_id** CFI function has the following input parameters:

Table 70: Input parameters of xd_free_zone_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
zone_ids	char**	-	Zone Id. list	-	-

7.31.4.Output parameters

This function does not return any value nor parameters.

7.32.xd_read_station

7.32.1.Overview

The **xd_read_station** CFI function reads the data of a station from a station database file.

7.32.2.Calling interface

The calling interface of the **xd_read_station** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name, station_id;
     xd_station_rec station_rec;
     long ierr[XD_NUM_ERR_READ_STATION];

     status = xd_read_station (file_name, station_id,
                               &station_rec, ierr);
}
```

7.32.3.Input parameters

The **xd_read_station** CFI function has the following input parameters:

Table 71: Input parameters of xd_read_station function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Station database file name	-	-
station_id	char*	-	Station name (Id)	-	-

7.32.4.Output parameters

The output parameters of the **xd_read_station** CFI function are:

Table 72: Output parameters of xd_read_station function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_station	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
station_rec	xd_station_rec	-	Station record data	-	-
ierr	long[]	-	Error vector	-	-

7.32.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_station** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_station** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 73: Error messages of xd_read_station function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Ground Station DB File not found.	No calculation performed	XD_CFI_READ_STATION_INIT_PARSER_ERR	0
ERR	Data Block not found.	No calculation performed	XD_CFI_READ_STATION_DATA_BLOCK_ERR	1
ERR	Data Block attribute not read.	No calculation performed	XD_CFI_READ_STATION_DATA_BLOCK_ATTRIBUTE_ERR	2
ERR	Data Block not of XML type.	No calculation performed	XD_CFI_READ_STATION_XML_TYPE_ERR	3
ERR	List_of_Ground_Stations not found	No calculation performed	XD_CFI_READ_STATION_LIST_GS_READ_ERR	4
ERR	Number of ground stations negative.	No calculation performed	XD_CFI_READ_STATION_LIST_GS_SIZE_ERR	5
ERR	Internal error returned.	No calculation performed	XD_CFI_READ_STATION_INTERNAL_1_ERR	6

ERR	Cannot read Station_Id.	No calculation performed	XD_CFI_READ_STATION_STATION_ID_READ_ERR	7
ERR	Station id not found.	No calculation performed	XD_CFI_READ_STATION_STATION_ID_NOT_FOUND_ERR	8
ERR	Error reading station record	No calculation performed	XD_CFI_READ_STATION_REC_READ_ERR	9

7.33.xd_read_station_file

7.33.1.Overview

The **xd_read_station_file** CFI function reads a whole station file for Earth Observation Missions.

7.33.2.Calling interface

The calling interface of the **xd_read_station_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_station_file station_data;
     long ierr[XD_NUM_ERR_READ_];

     status = xd_read_station_file (file_name,
                               &station_data, ierr);
}
```

7.33.3.Input parameters

The **xd_read_station_file** CFI function has the following input parameters:

Table 74: Input parameters of xd_read_station_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Station database file name	-	-

7.33.4.Output parameters

The output parameters of the **xd_read_station_file** CFI function are:

Table 75: Output parameters of xd_read_station_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_station_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
station_data	xd_station_file	-	Station file data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *station_data* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_station_file**.

7.33.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_station_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_station_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 76: Error messages of xd_read_station_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Ground Station DB File not found.	No calculation performed	XD_CFI_READ_STATION_FILE_INIT_PARSER_ERR	0
ERR	Data Block not found.	No calculation performed	XD_CFI_READ_STATION_FILE_DATA_BLOCK_ERR	1
ERR	Data Block attribute not read.	No calculation performed	XD_CFI_READ_STATION_FILE_DATA_BLOCK_ATTRIBUTE_ERR	2
ERR	Data Block not of XML type.	No calculation performed	XD_CFI_READ_STATION_FILE_XML_TYPE_ERR	3
ERR	List_of_Ground_Stations not found.	No calculation performed	XD_CFI_READ_STATION_FILE_LIST_GS_READ_ER R	4

ERR	Number of ground stations negative.	No calculation performed	XD_CFI_READ_STATION_FILE_LIST_GS_SIZE_ERR	5
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_STATION_FILE_MEM_ERR	6
ERR	Error reading station record number %d	No calculation performed	XD_CFI_READ_STATION_FILE_REC_READ_ERR	7

7.34.xd_free_station_file

7.34.1.Overview

The **xd_free_station_file** CFI function frees the memory allocated during the reading function xd_read_station_file.

7.34.2.Calling interface

The calling interface of the **xd_free_station_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_station_file station_data;
     xd_free_station_file (&station_data);
}
```

7.34.3.Input parameters

The **xd_free_station_file** CFI function has the following input parameters:

Table 77: Input parameters of xd_free_station_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
station_data	xd_station_file	-	Station file data structure	-	-

7.34.4.Output parameters

This function does not return any value nor parameters.

7.35. xd_read_station_id

7.35.1.Overview

The **xd_read_station_id** CFI function reads the list of station names (Id) contained in a station database file.

7.35.2.Calling interface

The calling interface of the **xd_read_station_id** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status, num_stations;
     char *file_name;
     char **station_list;
     long ierr[XD_NUM_ERR_READ_STATION_ID];

     status = xd_read_station_id (file_name, &num_stations,
                                  &station_list, ierr);
}
```

7.35.3.Input parameters

The **xd_read_station_id** CFI function has the following input parameters:

Table 78: Input parameters of xd_read_station_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Station database file name	-	-

7.35.4.Output parameters

The output parameters of the **xd_read_station_id** CFI function are:

Table 79: Output parameters of xd_read_station_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_station_id	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
num_stations	long	-	Number of stations	-	-
station_list	char**	.	Station list name	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *station_list* is a double pointer to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_station_id**.

7.35.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_station_id** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_station_id** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 80: Error messages of xd_read_station_id function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Ground Station DB File not found.	No calculation performed	XD_CFI_READ_STATION_ID_INIT_PARSER_ERR	0
ERR	Data Block not found.	No calculation performed	XD_CFI_READ_STATION_ID_DATA_BLOCK_ERR	1
ERR	List_of_Ground_Stations not found.	No calculation performed	XD_CFI_READ_STATION_ID_LIST_GS_READ_ERR	2
ERR	Number of ground stations negative.	No calculation performed	XD_CFI_READ_STATION_ID_LIST_GS_SIZE_ERR	3
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_STATION_ID_MEM_ERR	4
ERR	Error reading station Id.	No calculation performed	XD_CFI_READ_STATION_ID_READ_ID_ERR	5

7.36.xd_free_station_id

7.36.1.Overview

The **xd_free_station_id** CFI function frees the memory allocated during the reading function xd_read_station_id.

7.36.2.Calling interface

The calling interface of the **xd_free_station_id** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     char **station_ids;
     xd_free_station_id (&station_ids);
}
```

7.36.3.Input parameters

The **xd_free_station_id** CFI function has the following input parameters:

Table 81: Input parameters of xd_free_station_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
station_ids	char **	-	Station Id list	-	-

7.36.4. Output parameters

This function does not return any value nor parameters.

7.37. xd_read_star

7.37.1. Overview

The **xd_read_star** CFI function reads the data for a star from a star database file.

7.37.2. Calling interface

The calling interface of the **xd_read_star** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name, star_id;
     xd_star_rec star_data;
     long ierr[XD_NUM_ERR_READ_STAR];

     status = xd_read_star (file_name, star_id, &star_data, ierr);
}
```

7.37.3. Input parameters

The **xd_read_star** CFI function has the following input parameters:

Table 82: Input parameters of xd_read_star function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Star database file name	-	-
star_id	char*	-	Star name (Id) to be read	-	-

7.37.4. Output parameters

The output parameters of the **xd_read_star** CFI function are:

Table 83: Output parameters of xd_read_star function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_star	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
star_data	xd_star_rec	-	Star data structure	-	-
ierr	long[]	-	Error vector	-	-

7.37.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_star** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_star** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 84: Error messages of xd_read_star function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Star database file not found: %s	No calculation performed	XD_CFI_READ_STAR_FILE_NOT_FOUND_ERR	0
ERR	star id. %s not found in the star database file	No calculation performed	XD_CFI_READ_STAR_STAR_NOT_FOUND_ERR	1

7.38. xd_read_star_file

7.38.1.Overview

The **xd_read_star_file** CFI function reads a star database file for Earth Observation Missions.

7.38.2.Calling interface

The calling interface of the **xd_read_star_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_star_file star_data;
     long ierr[XD_NUM_ERR_READ_STAR_FILE];

     status = xd_read_star_file (file_name, &star_data, ierr);
}
```

7.38.3.Input parameters

The **xd_read_star_file** CFI function has the following input parameters:

Table 85: Input parameters of xd_read_star_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Star database file name (full path)	-	-

7.38.4.Output parameters

The output parameters of the **xd_read_star_file** CFI function are:

Table 86: Output parameters of xd_read_star_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_star_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
star_data	xd_star_file	-	Star file structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *star_data* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_star_file**.

7.38.5. Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_star_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_star_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 87: Error messages of xd_read_star_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the Star database file: %s	No calculation performed	XD_CFI_READ_STAR_FILE_FILE_NOT_FOUND_ER R	0
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_STAR_FILE_MEMORY_ERR	1
ERR	No stars found in file	No calculation performed	XD_CFI_READ_STAR_FILE_NO_STARS_ERR	2

7.39.xd_read_star_id

7.39.1.Overview

The **xd_read_star_id** CFI function reads the list of star names from star database files.

7.39.2.Calling interface

The calling interface of the **xd_read_star_id** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     char **star_list;
     long num_stars;
     long ierr[XD_NUM_ERR_READ_STAR_ID];

     status = xd_read_star_id (file_name, &num_stars,
                               &star_list, ierr);
}
```

7.39.3.Input parameters

The **xd_read_star_id** CFI function has the following input parameters:

Table 88: Input parameters of xd_read_star_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	Star database file	-	-

7.39.4.Output parameters

The output parameters of the **xd_read_star_id** CFI function are:

Table 89: Output parameters of xd_read_star_id function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_star_id	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
num_stars	long	-	Number of stars in the file	-	> 0
star_list	char**	-	Array of star names	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *star_list* is a double pointer to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_star_id**.

7.39.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_star_id** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_star_id** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 90: Error messages of xd_read_star_id function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the Star database file: %s	No calculation performed	XD_CFI_READ_STAR_ID_FILE_NOT_FOUND_ERR	0
ERR	Error allocating memory	No calculation performed	XD_CFI_READ_STAR_ID_MEMORY_ERR	1
ERR	No stars found in file	No calculation performed	XD_CFI_READ_STAR_ID_NO_STARS_ERR	2

7.40. xd_read_tle

7.40.1.Overview

The **xd_read_tle** CFI function read a TLE file.

7.40.2.Calling interface

The calling interface of the **xd_read_tle** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name, satellite;
     xd_tle_file tle_data;
     long ierr[XD_NUM_ERR_READ_TLE];

     status = xd_read_tle(file_name, satellite, &tle_data, ierr);
}
```

7.40.3.Input parameters

The **xd_read_tle** CFI function has the following input parameters:

Table 91: Input parameters of xd_read_tle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the orbit file.	-	-
satellite	char*	-	Satellite name as it appears in line 0 for a TLE. If it is an empty string (“”) or NULL, all the TLE are read, other way only the TLE for this satellite are read.	-	-

7.40.4.Output parameters

The output parameters of the **xd_read_tle** CFI function are:

Table 92: Output parameters of xd_read_tle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_tle	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
tle_data	xd_tle_file	-	Orbital state vectors data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *tle_data* is a pointer to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_tle**.

7.40.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_tle** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_tle** function by callingthe function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 93: Error messages of xd_read_tle function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the TLE file %s	File not read	XD_CFI_READ_TLE_FILE_NOT_FOUND_ERR	0
ERR	Wrong file format %s, line 0	File not read	XD_CFI_READ_TLE_WRONG_LINE0_ERR	1
ERR	Wrong file format %s, line 1	File not read	XD_CFI_READ_TLE_WRONG_LINE1_ERR	2
ERR	Wrong file format %s, line 2	File not read	XD_CFI_READ_TLE_WRONG_LINE2_ERR	3
ERR	Error allocating memory	File not read	XD_CFI_READ_TLE_MEM_ERR	4
ERR	Wrong file format %s. Satellite number in line 1 and 2 should be equal	File not read	XD_CFI_READ_TLE_WRONG_SAT_ERR	5

ERR	No TLE found in %s	No TLE read File not read	XD_CFI_READ_TLE_NO_LINES_ERR	6
WARN	Wrong file format %s, line 1. Wrong checksum value. TLE discarded	TLE skipped	XD_CFI_READ_TLE_WRONG_CHECKSUM1_WARN	7
WARN	Wrong file format %s, line 2. Wrong checksum value. TLE discarded	TLE skipped	XD_CFI_READ_TLE_WRONG_CHECKSUM2_WARN	8

7.41. xd_free_tle

7.41.1.Overview

The **xd_free_tle** CFI function frees the memory allocated during the reading function **xd_read_tle**.

7.41.2.Calling interface

The calling interface of the **xd_free_tle** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_tle_file tle_data;
     xd_free_tle (&tle_data);
}
```

7.41.3.Input parameters

The **xd_free_tle** CFI function has the following input parameters:

Table 94: Input parameters of xd_free_tle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
tle_data	xd_tle_file	-	TLE data that has been read with xd_read_tle	-	-

7.41.4.Output parameters

This function does not return any value nor parameters.

7.42. xd_read_precise_propag_file

7.42.1. Overview

The **xd_read_precise_propag_file** CFI function read a configuration file for precise propagation.

7.42.2. Calling interface

The calling interface of the **xd_read_precise_propag_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_propag_precise_config precise_conf;
     long ierr[XD_NUM_ERR_READ_PRECISE_PROPAG];

     status = xd_read_precise_propag_file(file_name,
                                          &precise_conf, ierr);
}
```

7.42.3. Input parameters

The **xd_read_precise_propag** CFI function has the following input parameters:

Table 95: Input parameters of xd_read_precise_propag function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the orbit file.	-	-

7.42.4. Output parameters

The output parameters of the **xd_read_precise_propag** CFI function are:

Table 96: Output parameters of xd_read_precise_propag function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_precise_propag	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
precise_conf	xd_propag_precise_config	-	Structure that will contain the precise configuration data for precise propagation.	-	-
ierr	long[]	-	Error vector	-	-

7.42.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_read_precise_propag** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_read_precise_propag** function bycalling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 97: Error messages of xd_read_precise_propag function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open file	File not read	XD_CFI_READ_PRECISE_PROPAG_INIT_PARSER_ERR	0
ERR	Could not read parameter %s	File not read	XD_CFI_READ_PRECISE_PROPAG_READ_PARAM_ERR	1
ERR	Flag nor correct. Its value must be 0 or 1	File not read	XD_CFI_READ_PRECISE_PROPAG_WRONG_FLAG_ERR	2
ERR	Could not close the file	File not read	XD_CFI_READ_PRECISE_PROPAG_CLEANUP_PARSER_ERR	3
ERR	Could not write the fixed header	File not read	XD_CFI_WRITE_PRECISE_PROPAG_WRITE_FHR_ERR	4
WARN	Cannot write schema in the file		XD_CFI_WRITE_PRECISE_PROPAG_SET_SCHEMA_WARN	5

7.43. xd_write_orbit_file

7.43.1. Overview

The **xd_write_orbit_file** CFI function writes an orbit file in XML format using the data structure provided by the user. The orbit file can be either:

- A Predicted orbit file
- A Restituted orbit file
- A DORIS Predicted file
- The Time_Reference and Ref_Frame fields in the variable header of the orbit file are filled according to the parameters time_ref_of and ref_frame in the OSV records. Therefore it is required that all OSVs contained in xd_orbit_file have the same time reference and reference frame.

7.43.2. Calling interface

The calling interface of the **xd_write_orbit_file** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_fhr fhr;
     xd_orbit_file *osv_data;
     long ierr[XD_NUM_ERR_WRITE_ORBIT_FILE];

     status = xd_write_orbit_file(file_name, &fhr, &osv_data, ierr);
}
```

7.43.3. Input parameters

The **xd_write_orbit_file** CFI function has the following input parameters:

Table 98: Input parameters of xd_write_orbit_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the orbit file. If empty string (i.e, ""), then the file is written with the name in the fixed_header structure (fhr)	-	-
fhr	xd_fhr	-	Fixed header structure	-	-
xd_orbit_file	osv_data	-	Orbital state vectors data structure	-	-

7.43.4.Output parameters

The output parameters of the **xd_write_orbit_file** CFI function are:

Table 99: Output parameters of xd_write_orbit_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_write_orbit_file	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
ierr	long[]	-	Error vector	-	-

7.43.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_write_orbit_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_write_orbit_file** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 100: Error messages of xd_write_orbit_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot create root element	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_CREATE_TREE_ERR	0
ERR	Cannot create in-memory XML tree	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_CREATE_ROOT_ERR	1
ERR	Cannot write the fixed header	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_WRITE_FHR_ERR	2
ERR	Cannot add XML node to tree: %s	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_CREATE_NODE_ERR	3
ERR	Cannot convert time from processing to external	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_GET_ASCII_TIME_ERR	4
ERR	Cannot write XML file	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_WRITE_ERR	5
ERR	Cannot go to the desired node	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_GOTO_NODE_ERR	6

WARN	Cannot write schema in the file	File written to disk but without schema	XD_CFI_WRITE_ORBIT_FILE_SET_SCHEMA_WARN	7
ERR	All the orbit records must have the same reference frame	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_REF_FRAME_ERR	8
ERR	All the orbit records must have the same time reference	No calculation performed	XD_CFI_WRITE_ORBIT_FILE_TIME_REF_OF_ERR	9

7.44. xd_write_osf

7.44.1.Overview

The **xd_write_osf** CFI function writes an Orbit Scenario file in XML format using the data provided by the user.

7.44.2.Calling interface

The calling interface of the **xd_write_osf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_fhr fhr;
     xd_osf_file osf_data;
     long ierr[XD_NUM_ERR_WRITE_OSF];

     status = xd_write_osf (file_name, &fhr, &osf_data, ierr);
}
```

7.44.3.Input parameters

The **xd_write_osf** CFI function has the following input parameters:

Table 101: Input parameters of xd_write_osf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the orbit scenario file. If empty string (i.e, ““), then the file is written with the name in the fixed_header structure (fhr)	-	-
fhr	xd_fhr	-	Fixed header structure	-	-
xd_osf_file	osf_data	-	Orbital changes data structure	-	-

7.44.4.Output parameters

The output parameters of the **xd_write_osf** CFI function are:

Table 102: Output parameters of xd_write_osf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_write_osf	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
ierr	long[]	-	Error vector	-	-

7.44.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_write_osf** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_write_osf** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 103: Error messages of xd_write_osf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot create in-memory XML tree	No calculation performed	XD_CFI_WRITE_OSF_CREATE_TREE_ERR	0
ERR	Cannot write the fixed header	No calculation performed	XD_CFI_WRITE_OSF_WRITE_FHR_ERR	1
ERR	Cannot create root element	No calculation performed	XD_CFI_WRITE_OSF_CREATE_ROOT_ERR	2
ERR	Cannot add XML node to tree	No calculation performed	XD_CFI_WRITE_OSF_CREATE_NODE_ERR	3
ERR	Cannot set XML node value	No calculation performed	XD_CFI_WRITE_OSF_SET_NODE_VALUE_ERR	4
ERR	Cannot convert time from processing to external	No calculation performed	XD_CFI_WRITE_OSF_TIME_TO_EXTERNAL_ERR	5
ERR	Cannot write XML file	No calculation performed	XD_CFI_WRITE_OSF_WRITE_ERR	6
WARN	Cannot write schema in the file	File written to disk but without schema	XD_CFI_WRITE_OSF_SET_SCHEMA_WARN	7
ERR	Time reference of orbital changes must be UT1	No calculation performed	XD_CFI_WRITE_OSF_TIME_REF_OF_ERR	8

7.45. xd_write_doris

7.45.1. Overview

The **xd_write_doris** CFI function writes a DORIS NAVIGATOR Product file for CRYOSAT, using the data provided by the user.

7.45.2. Calling interface

The calling interface of the **xd_write_doris** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_doris_mph_sph fhr;
     xd_doris_file doris_data;
     long ierr[XD_NUM_ERR_WRITE_DORIS];

     status = xd_write_doris (file_name, &fhr, &doris_data, ierr);
}
```

7.45.3. Input parameters

The **xd_write_doris** CFI function has the following input parameters:

Table 104: Input parameters of xd_write_doris function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	DORIS file name	-	-
fhr	xd_doris_mph_sph	-	Main and Specific product headers	-	-
doris_data	xd_doris_file	-	DORIS data structure	-	-

7.45.4. Output parameters

The output parameters of the **xd_write_doris** CFI function are:

Table 105: Output parameters of xd_write_doris function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_write_doris	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
ierr	long[]	-	Error vector	-	-

7.45.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_write_doris** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_write_doris** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 106: Error messages of xd_write_doris function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the file %s for writing	No calculation performed	XD_CFI_WRITE_DORIS_OPEN_ERR	0
ERR	Error writing the fixed header	No calculation performed	XD_CFI_WRITE_DORIS_WRITE_FHR_ERR	1
ERR	Error writing the binary data	No calculation performed	XD_CFI_WRITE_DORIS_WRITE_BINARY_ERR	2

7.46.xd_write_stf

7.46.1.Overview

The **xd_write_stf** CFI function writes a swath template file XML format using the data provided by the user.

7.46.2.Calling interface

The calling interface of the **xd_write_stf** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
      long status;
      char *file_name;
      xd_fhr fhr;
      xd_stf_file stf_data;
      long ierr[XD_NUM_ERR_WRITE_STF];

      status = xd_write_stf (file_name, &fhr, &stf_data, ierr);
}
```

7.46.3.Input parameters

The **xd_write_stf** CFI function has the following input parameters:

Table 107: Input parameters of xd_write_stf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the swath template file. If empty string (i.e, ““), then the file is written with the name in the fixed_header structure (fhr)	-	-
fhr	xd_fhr	-	Fixed header structure	-	-
xd_stf_file	stf_data	-	STF data structure	-	-

7.46.4.Output parameters

The output parameters of the **xd_write_stf** CFI function are:

Table 108: Output parameters of xd_write_stf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_write_stf	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
ierr	long[]	-	Error vector	-	-

7.46.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_write_stf** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_write_stf** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 109: Error messages of xd_write_stf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot create XML tree.	No calculation performed	XD_CFI_WRITE_STF_CREATE_XML_ERR	0
ERR	Cannot create root node in the XML tree.	No calculation performed	XD_CFI_WRITE_STF_CREATE_ROOT_XML_ERR	1
ERR	Error writing fixed header.	No calculation performed	XD_CFI_WRITE_STF_XD_FHR_WRITE_ERR	2
ERR	Error while writing Swath Template File variable header.	No calculation performed	XD_CFI_WRITE_STF_XD_STF_VHR_WRITE_ERR	3
ERR	Cannot create the node %s	No calculation performed	XD_CFI_WRITE_STF_CREATE_NODE_ERR	4
ERR	Wrong swath_type	No calculation performed	XD_CFI_WRITE_STF_WRONG_SWATH_TYPE_ERR	5
ERR	Error while writing the swath record n.%d	No calculation performed	XD_CFI_WRITE_STF_WRITE_REC_ERR	6
ERR	Cannot write to disk the XML tree	No calculation performed	XD_CFI_WRITE_STF_WRITE_ERR	7
WARN	Cannot write schema in the file	File written to disk but without schema	XD_CFI_WRITE_STF_SET_SCHEMA_WARN	8

7.47.xd_write_att

7.47.1.Overview

The **xd_write_att** CFI function writes an attitude generic file in XML format using the data provided by the user.

Note about output format: the number of decimal digits written to file depends on the type of data:

- If angles are used, 6 decimal digits are written.
- If quaternions are used, 9 decimal digits are written.

It is done this way because having 9 decimal digits in quaternions reduces pointing error significantly .

7.47.2.Calling interface

The calling interface of the **xd_write_att** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_fhr fhr;
     xd_att_file att_data;
     long ierr[XD_NUM_ERR_WRITE_ATT];

     status = xd_write_att (file_name, &fhr, &att_data, ierr);
}
```

7.47.3.Input parameters

The **xd_write_att** CFI function has the following input parameters:

Table 110: Input parameters of xd_write_att function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the attitude file. If empty string (i.e, ""), then the file is written with the name in the fixed_header structure (fhr)	-	-
fhr	xd_fhr	-	Fixed header structure	-	-
xd_att_file	att_data	-	Attitude data structure	-	-

7.47.4. Output parameters

The output parameters of the **xd_write_att** CFI function are:

Table 111: Output parameters of xd_write_att function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_write_att	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
ierr	long[]	-	Error vector	-	-

7.47.5. Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_write_att** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_write_att** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 112: Error messages of xd_write_att function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot create in-memory XML tree	No calculation performed	XD_CFI_WRITE_ATT_CREATE_TREE_ERR	0
ERR	Cannot create root element	No calculation performed	XD_CFI_WRITE_ATT_CREATE_ROOT_ERR	1
ERR	Cannot write the fixed header	No calculation performed	XD_CFI_WRITE_ATT_WRITE_FHR_ERR	2
ERR	Cannot add XML node to tree: %s	No calculation performed	XD_CFI_WRITE_ATT_CREATE_NODE_ERR	3
ERR	Cannot convert time from processing to external	No calculation performed	XD_CFI_WRITE_ATT_GET_ASCII_TIME_ERR	4
ERR	Cannot go to the desired node	No calculation performed	XD_CFI_WRITE_ATT_GOTO_NODE_ERR	5
ERR	Cannot write XML file	No calculation performed	XD_CFI_WRITE_ATT_WRITE_ERR	6

WARN	Cannot write schema in the file	File written to disk but without schema	XD_CFI_WRITE_ATT_SET _SCHEMA_WARN	7

7.48.xd_write_tle

7.48.1.Overview

The **xd_write_tle** CFI function writes a TLE file. The data to be written are in the input structure except for the checksum, that it is computed for every line.

7.48.2.Calling interface

The calling interface of the **xd_write_tle** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status;
     char *file_name;
     xd_tle_file tle_data;
     long ierr[XD_NUM_ERR_WRITE_TLE]

     status = xd_write_tle (file_name, &tle_data, ierr);
}
```

7.48.3.Input parameters

The **xd_write_tle** CFI function has the following input parameters:

Table 113: Input parameters of xd_write_tle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_name	char*	-	File name for the TLE file.	-	-
xd_tle_file	tle_data	-	TLE data structure	-	-

7.48.4.Output parameters

The output parameters of the **xd_write_tle** CFI function are:

Table 114: Output parameters of xd_write_tle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range

xd_write_tle	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
ierr	long[]	-	Error vector	-	-

7.48.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_write_tle** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_write_tle** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 115: Error messages of xd_write_tle function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not open the TLE file for writing: %s	No calculation performed	XD_WRITE_TLE_FILE_OPEN_ERR	0
ERR	Could not write the TLE file: %s	No calculation performed	XD_WRITE_TLE_WRITE_ERR	1

7.49.xd_xml_validate

7.49.1.Overview

The **xd_xml_validate** CFI function validates an XML file using its XML schema and checks the XML schema versioning.

7.49.2.Calling interface

The calling interface of the CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     long status, valid_status;
     char *filename, *schema, *logfile;
     long mode;
     long ierr[XD_NUM_ERR_XML_VALIDATE];

     status = xd_xml_validate (filename, &mode, schema, logfile,
                               &valid_status, ierr);
}
```

7.49.3.Input parameters

The **xd_xml_validate** CFI function has the following input parameters:

Table 116: Input parameters of xd_xml_validate function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
filename	char*	-	File name to validate	-	-
mode	long	-	Flag to select the schema to be used to validate the file. It can be either: • XD_DEFAULT_SCHEMA: use the schema that is in the root element of the XML file. or • XD_USER_SCHEMA: use the schema given in the *schema* parameter in the interface.	-	-
schema	char*	-	Schema file. The schema can be given as an absolute path or as a relative path from the file's directory (No the current directory)	-	-

logfile	char*	-	Log file (file path). It is used to store the messages returned by the validation process. The result of the validation can be seen at the end of the log in the following message: Validation result for "filename": [VALID]/[INVALID]	-	-

7.49.4.Output parameters

The output parameters of the **xd_xml_validate** CFI function are:

Table 117: Output parameters of xd_xml_validate function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_xml_validate	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
valid_status	long	-	The result of the validation: • XD_XML_INVALID (= -1) • XD_XML_VALID (= 0)	-	-
ierr	long[]	-	Error vector	-	-

7.49.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_xml_validate** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_xml_validate** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 118: Error messages of xd_xml_validate function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Could not set schema within the XML file Severe errors in the file format.	The file is not well formed and cannot be opened because of severe errors. No calculation performed	XD_CFI_XML_VALIDATE_SET_SCHEMA_ERR	0

ERR	Could not open file: %s. Severe errors in the file format	The file is not well formed and cannot be opened because of severe errors. No calculation performed	XD_CFI_XML_VALIDATE_INIT_PARSER_ERR	1
ERR	Memory allocation error	No calculation performed	XD_CFI_XML_VALIDATE_MEMORY_ERR	2
ERR	No schema provided	No calculation performed	XD_CFI_XML_VALIDATE_NO_SCHEMA_ERR	3
ERR	Wrong input mode	No calculation performed	XD_CFI_XML_VALIDATE_WRONG_MODE_ERR	4
ERR	Could not open file: %s	No calculation performed	XD_CFI_XML_VALIDATE_OPEN_FILE_ERR	5
ERR	Could not copy input file to the current directory	No calculation performed	XD_CFI_XML_VALIDATE_COPY_FILE_ERR	6
ERR	Schema not found in root element	No calculation performed	XD_CFI_XML_VALIDATE_NO_SCHEMA_IN_FILE_ERR	7
ERR	Schema version differs from the version in the schema filename	No calculation performed	XD_CFI_XML_VALIDATE_INCONSITENT_SCHEMA_VERS_ERR	8
WARN	The XML file does not contain the schema version	Calculation performed	XD_CFI_XML_VALIDATE_NO_SCH_VERS_IN_FILE_WARN	9
WARN	Schema version not found	Calculation performed	XD_CFI_XML_VALIDATE_NO_VERS_IN_SCHEMA_WARN	10
WARN	Schema version in XML file is older than the schema version	Calculation performed	XD_CFI_XML_VALIDATE_LESS_SCHEMA_VERS_WARN	11
WARN	Schema version in XML file is newer than the schema version	Calculation performed	XD_CFI_XML_VALIDATE_GREATER_SCHEMA_VERS_WARN	12

7.49.6.Executable program

An XML file can also be validated using the executable program **xml_validate.** It can be called from a Unix shell as:

```
xml_validate -file filename
[-sch schema_filename] [-log log_filename]
[ -help ] [ -v ]
[ -show ]
```

Note that:

- Order of parameters does not matter.
- Bracketed parameters are not mandatory.
- [-v] option for Verbose mode (default is Silent).

- [-show] displays the inputs of the function and the results.
- The filename is validated using the schema_filename if it is provided. If not, the default schema is used (the one in the root element of the file).
- The validation log is stored in the log_filename. By default the standard output is used.

Example:

```
xml_validate -file ../../data/CRYOSAT_XML_OSF
-sch ../../../../schemas/public/CS_OPER_MPL_ORBSCT_01.00.XSD
-log log_file_exe -show
```

7.50.xd_select_schema

7.50.1.Overview

The **xd_select_schema** returns the most recent schema file name applicable for a given file type and mission.

7.50.2.Calling interface

The calling interface of the CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_fileinfo info;
     char        schema[XD_MAX_STR];
     long        ierr[XD_NUM_ERR_SELECT_SCHEMA];

     status = xd_select_schema(&info, schema,
                                ierr);
}
```

7.50.3.Input parameters

The **xd_select_schema** CFI function has the following input parameters:

Table 119: Input parameters of xd_select_schema function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
info	xd_fileinfo	-	File info containing the mission and the file type (see Table 3)	-	-

7.50.4.Output parameters

The output parameters of the **xd_select_schema** CFI function are:

Table 120: Output parameters of xd_select_schema function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_select_schema	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
schema	char*	-	Schema name	-	-
ierr	long[]	-	Error vector	-	-

7.50.5.Warnings and errors

The current version of the **xd_select_schema** does not return any errors nor warnings.

7.51.xd_orbit_file_decimate

7.51.1.Overview

The **xd_orbit_file _decimate** adds capability to configure position interpolator according to user need (decimation).

The decimation is performed in the orbit file structure. This way user has two options using the output of this function:

1.to write a new orbit file and use this file to initialize the orbit id.

2.To initialize directly the orbit id with the new structure.

The fucntion works as follows:

•First and last state vectors in input list are copied to output list.

•Using the input decimation delta (D), and being t0 the time of the first state vector of the input list, the state vectors whose time is closer to time t=t0+k*D (k = 1, 2...n, t0<t<tn) are copied to output list.

7.51.2.Calling interface

The calling interface of the **xd_orbit_file_decimate** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_fhr fhr_in, fhr_out;
     xd_orbit_file osv_in, osv_out;
     double decimation_delta_time;
     long ierr[XD_NUM_ERR_ORBIT_FILE_DECIMATE];
     status = xd_orbit_file_decimate(&fhr_in, &osv_in,
                                     decimation_delta_time,
                                     &fhr_out, &osv_out,
                                     ierr);
}
```

7.51.3.Input parameters

The **xd_orbit_file_decimate** CFI function has the following input parameters:

Table 121: Input parameters of xd_orbit_file_decimate function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range

fhr_in	xd_fhr	-	Data structure containing the fixed header data read from the input file	-	-
osv_in	xd_orbit_file	-	Data structure containing the data read from the input file	-	-
decimation_delta_time	double	-	Delta time used for decimation process.	seconds	>=0.

7.51.4. Output parameters

The output parameters of the **xd_orbit_file_decimate** CFI function are:

Table 122: Output parameters of xd_orbit_file_decimate function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_orbit_file_decimate	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
fhr_out	xd_fhr	-	Data structure containing the fixed header for output file		
osv_out	xd_orbit_file	-	Data structure containing the output file data	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *osv_out* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_orbit_file**

7.51.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_orbit_file_decimate** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_orbit_file_decimate** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 123: Error messages of xd_orbit_file_decimate function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error allocating memory	No calculation performed	XD_ORBIT_FILE_DECIMATE_MEM_ERR	0
WARN	The time difference between 2 consecutive OSVs is greater than twice input time decimation delta	No calculation performed	XD_ORBIT_FILE_DECIMATE_DELTA_WARN	1
ERR	Error computing validity interval	No calculation performed	XD_ORBIT_FILE_DECIMATE_VAL_TIME_ERR	2

7.52.xd_attitude_file_decimate

7.52.1.Overview

The **xd_attitude_file _decimate** adds capability to configure attitude interpolator according to user need (decimation).

The decimation is performed in the attitude file structure. This way user has two options using the output of this function:

1.to write a new attitude file and use this file to initialize the attitude id.

2.To initialize directly the attitude id with the new structure.

The fucntion works as follows:

•First and last attitude records in input list are copied to output list.

•Using the input decimation delta (D), and being t0 the time of the first attitude record of the input list, the attitude records whose time is closer to time t=t0+k*D (k = 1, 2...n, t0<t<tn) are copied to output list.

7.52.2.Calling interface

The calling interface of the **xd_attitude_file_decimate** CFI function is the following (input parameters are underlined):

```
#include <explorer_data_handling.h>
{
     xd_fhr fhr_in, fhr_out;
     xd_att_file att_in, att_out;
     double decimation_delta_time;
     long ierr[XD_NUM_ERR_ATTITUDE_FILE_DECIMATE];
     status = xd_attitude_file_decimate(&fhr_in, &att_in,
                                       decimation_delta_time,
                                       &fhr_out, &att_out,
                                       ierr);
}
```

7.52.3.Input parameters

The **xd_attitude_file_decimate** CFI function has the following input parameters:

Table 124: Input parameters of xd_attitude_file_decimate function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range

fhr_in	xd_fhr	-	Data structure containing the fixed header data read from the input file	-	-
att_in	xd_att_file	-	Data structure containing the data read from the input file	-	-
decimation_delta_time	double	-	Delta time used for decimation process.	seconds	>=0.

7.52.4. Output parameters

The output parameters of the **xd_attitude_file_decimate** CFI function are:

Table 125: Output parameters of xd_attitude_file_decimate function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_attitude_file_decimate	long	-	Function status flag: • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated	-	-
fhr_out	xd_fhr	-	Data structure containing the fixed header for output file		
att_out	xd_att_file	-	Data structure containing the output file data	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The *osv_out* structure contains pointers to memory allocated dinamically. In order to avoid memory leaks, the user will have to free that memory when the data structure is not to be used any more. The memory can be freed by calling to the CFI function **xd_free_att.**

7.52.5.Warnings and errors

Next table lists the possible error messages that can be returned by the **xd_attitude_file_decimate** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_DATA_HANDLING software library **xd_get_msg** (see [GEN_SUM]).

This table also indicates the type of message returned, i.e. either a warning (`WARN`) or an error (`ERR`), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the error vector returned by the **xd_attitude_file_decimate** function by calling the function of the EO_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM])

Table 126: Error messages of xd_attitude_file_decimate function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error allocating memory	No calculation performed	XD_CFI_ATTITUDE_FILE_DECIMATE_MEM_ERR	0
WARN	The time difference between 2 consecutive records is greater than twice input time decimation delta	Calculation performed	XD_CFI_ATTITUDE_FILE_DECIMATE_DELTA_WARN	1
ERR	Error computing validity interval	No calculation performed	XD_CFI_ATTITUDE_FILE_DECIMATE_VAL_TIME_ERR	2
WARN	Attitude record reference not UTC. UTC Validity interval computed extending one minute or more (rounded to have exact number of minutes) every end of interval	Calculation performed	XD_CFI_ATTITUDE_FILE_DECIMATE_ATT_NOT_UTC_WARN	3

8. LIST OF SCHEMA'S VERSIONS

Next table shows the list of last version (default version) for the schema of the XML files that are handled by the CFI. The default schema depends on the mission, so different versions appear depending on the Mission Columt. For the sake of simplicity, in this column could appear:

- Sat_1: All satellites except SENTINEL missions and EarthCARE mission.
- Sat_2: SENTINEL missions and EarthCARE mission.

Table 127: Default schema's versions

File type	Mission	Version	Schema
Predicted Orbit files	Sat_1	1.4	EO_OPER_MPL_ORBPRE_0104.XSD
	Sat_2	2.1	EO_OPER_MPL_ORBPRE_0201.XSD
Restituted Orbit files	Sat_1	1.4	EO_OPER_AUX_ORBRES_0104.XSD
	Sat_2	2.1	EO_OPER_AUX_ORBRES_0201.XSD
Doris Preliminary files	Sat_1	1.4	EO_OPER_AUX_ORBDOP_0104.XSD
	Sat_2	2.1	EO_OPER_AUX_ORBDOP_0201.XSD
Doris Precise files	Sat_1	1.4	EO_OPER_AUX_ORBDOR_0104.XSD
	Sat_2	2.1	EO_OPER_AUX_ORBDOR_0201.XSD
Orbit Scenario files	Sat_1	1.4	EO_OPER_MPL_ORBSCT_0104.XSD
	Sat_2	2.2	EO_OPER_MPL_ORBSCT_0202.XSD
Orbit Event files	Sat_1	1.4	EO_OPER_MPL_ORBREF_0104.XSD
	Sat_2	2.1	EO_OPER_MPL_ORBREF_0201.XSD
DORIS Navigator files	Sat_1	1.2	EO_OPER_DOR_NAV_0_.HeaderTypes_0102.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD
	Sat_2	2.1	EO_OPER_DOR_NAV_0_.HeaderTypes_0201.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD
Star Tracker files	Sat_1	1.2	EO_OPER_STR1ATT_0_HeaderTypes_0102.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0101.XSD
	Sat_2	2.1	EO_OPER_STR1ATT_0_HeaderTypes_0201.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0100.XSD
Satellite Configuration File	Sat_1	1.3	EO_OPER_INT_SATCFG_0103.XSD
	Sat_2	2.1	EO_OPER_INT_SATCFG_0201.XSD
Attitude File	Sat_1	1.2	EO_OPER_INT_ATTREF_0102.XSD

File type	Mission	Version	Schema
	Sat_2	2.1	EO_OPER_INT_ATTREF_0201.XSD
Star tracker configuration File	Sat_1	1.2	EO_OPER_INT_STRCFG_0102.XSD
	Sat_2	2.1	EO_OPER_INT_STRCFG_0201.XSD
DEM Configuration File	Sat_1	1.6	EO_OPER_INT_DEMCFG_0106.XSD
	Sat_2	2.3	EO_OPER_INT_DEMCFG_0203.XSD
Swath Definition File	Sat_1	2.2	EO_OPER_MPL_SW_DEF_0202.XSD
	Sat_2	3.1	EO_OPER_MPL_SW_DEF_0301.XSD
Swath Template File	Sat_1	2.2	EO_OPER_MPL_SWTREF_0202.XSD
	Sat_2	3.1	EO_OPER_MPL_SWTREF_0301.XSD
Zone Database File	Sat_1	1.3	EO_OPER_MPL_ZON_DB_0103.XSD
	Sat_2	2.1	EO_OPER_MPL_ZON_DB_0201.XSD
Station Database File	Sat_1	1.5	EO_OPER_MPL_GND_DB_0105.XSD
	Sat_2	2.1	EO_OPER_MPL_GND_DB_0201.XSD
Precise Propagator Configuration File	Sat_1	1.1	EO_OPER_INT_PPRCFG_0101.XSD
	Sat_2	2.1	EO_OPER_INT_PPRCFG_0201.XSD

However the current CFI libraries are able to use older XML file versions. Next table shows all the available schema's versions for the Earth Observation Missions files at the time of the CFI's package delivery, and which of them are supported by the current CFI's implementation.

The meaning of the column “Supp. CFI's” is:

- N: Version not supported for reading/writing.
- Y: Supported version for reading and writing.
- R: Supported only for reading.

Note that the writing functions create files with the version labeled with “Y”.

Note also that for those files that can be written by the CFI, the written version depend on the satellite mission (For reading, the files can be read independently of the mission).

Table 128: List of schema's versions

File type	Version	Schema	Mission	Supp. CFI's
Predicted Orbit files	1.0	EO_OPER_MPL_ORBPRE_0100.XSD	-	N
	1.1	EO_OPER_MPL_ORBPRE_0101.XSD	-	R
	1.2	EO_OPER_MPL_ORBPRE_0102.XSD	-	R
	1.3	EO_OPER_MPL_ORBPRE_0103.XSD	-	R
	1.4	EO_OPER_MPL_ORBPRE_0104.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_MPL_ORBPRE_0200.XSD	-	R
	2.1	EO_OPER_MPL_ORBPRE_0201.XSD	Sat_1	N
			Sat_2	Y
Restituted Orbit files	1.0	EO_OPER_AUX_ORBRES_0100.XSD	-	N
	1.1	EO_OPER_AUX_ORBRES_0101.XSD	-	R
	1.2	EO_OPER_AUX_ORBRES_0102.XSD	-	R
	1.3	EO_OPER_AUX_ORBRES_0103.XSD	-	R
	1.4	EO_OPER_AUX_ORBRES_0104.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_AUX_ORBRES_0200.XSD		
	2.1	EO_OPER_AUX_ORBRES_0201.XSD	Sat_1	N
			Sat_2	Y
Doris Preliminary files	1.0	EO_OPER_AUX_ORBDOP_0100.XSD	-	N
	1.1	EO_OPER_AUX_ORBDOP_0101.XSD	-	R
	1.2	EO_OPER_AUX_ORBDOP_0102.XSD	-	R
	1.3	EO_OPER_AUX_ORBDOP_0103.XSD	-	R
	1.4	EO_OPER_AUX_ORBDOP_0104.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_AUX_ORBDOP_0200.XSD	-	R

File type	Version	Schema	Mission	Supp. CFI's
	2.1	EO_OPER_AUX_ORBDOP_0201.XSD	Sat_1	N
			Sat_2	Y
Doris Precise files	1.0	EO_OPER_AUX_ORBDOR_0100.XSD	-	N
	1.1	EO_OPER_AUX_ORBDOR_0101.XSD	-	R
	1.2	EO_OPER_AUX_ORBDOR_0102.XSD	-	R
	1.3	EO_OPER_AUX_ORBDOR_0103.XSD	-	R
	1.4	EO_OPER_AUX_ORBDOR_0104.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_AUX_ORBDOR_0200.XSD	-	R
	2.1	EO_OPER_AUX_ORBDOR_0201.XSD	Sat_1	N
			Sat_2	Y
Orbit Scenario files	1.0	EO_OPER_MPL_ORBSCT_0100.XSD	-	N
	1.1	EO_OPER_MPL_ORBSCT_0101.XSD	-	R
	1.2	EO_OPER_MPL_ORBSCT_0102.XSD	-	R
	1.3	EO_OPER_MPL_ORBSCT_0103.XSD	-	R
	1.4	EO_OPER_MPL_ORBSCT_0104.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_MPL_ORBSCT_0200.XSD	-	R
	2.1	EO_OPER_MPL_ORBSCT_0201.XSD	-	R
	2.2	EO_OPER_MPL_ORBSCT_0202.XSD	Sat_1	N
			Sat_2	Y
Orbit Event files	1.0	EO_OPER_MPL_ORBREF_0100.XSD	-	N
	1.1	EO_OPER_MPL_ORBREF_0101.XSD	-	R
	1.2	EO_OPER_MPL_ORBREF_0102.XSD	-	R
	1.3	EO_OPER_MPL_ORBREF_0103.XSD	-	R
	1.4	EO_OPER_MPL_ORBREF_0104.XSD	Sat_1	Y
			Sat_2	N

File type	Version	Schema	Mission	Supp. CFI's
	2.0	EO_OPER_MPL_ORBREF_0200.XSD	-	R
	2.1	EO_OPER_MPL_ORBREF_0201.XSD	Sat_1	N
			Sat_2	Y
DORIS Navigator files	1.1	EO_OPER_DOR_NAV_0_.HeaderTypes_0101.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD	-	R
	1.2	EO_OPER_DOR_NAV_0_.HeaderTypes_0102.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_DOR_NAV_0_.HeaderTypes_0200.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD	-	R
	2.1	EO_OPER_DOR_NAV_0_.HeaderTypes_0201.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD	Sat_1	N
			Sat_2	Y
Star Tracker files	1.0	EO_OPER_STR1ATT_0_HeaderTypes_0100.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0100.XSD	-	R
	1.1	EO_OPER_STR1ATT_0_HeaderTypes_0101.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0100.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_STR1ATT_0_HeaderTypes_0200.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0100.XSD	-	R
	2.1	EO_OPER_STR1ATT_0_HeaderTypes_0201.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0100.XSD	Sat_1	N
			Sat_2	Y
Satellite Configuration File	1.0	EO_OPER_INT_SATCFG_0100.XSD	-	N
	1.1	EO_OPER_INT_SATCFG_0101.XSD	-	N
	1.2	EO_OPER_INT_SATCFG_0102.XSD	-	R
	1.3	EO_OPER_INT_SATCFG_0103.XSD	-	R
	2.0	EO_OPER_INT_SATCFG_0200.XSD	-	R
	2.1	EO_OPER_INT_SATCFG_0201.XSD	-	R
Attitude File	1.0	EO_OPER_INT_ATTCFG_0100.XSD	-	N
	1.1	EO_OPER_INT_ATTREF_0101.XSD	-	R

File type	Version	Schema	Mission	Supp. CFI's
	1.2	EO_OPER_INT_ATTREF_0102.XSD	Sat_1	Y
			Sat_2	N
	2.0	EO_OPER_INT_ATTREF_0200.XSD	-	R
	2.1	EO_OPER_INT_ATTREF_0201.XSD	Sat_1	N
			Sat_2	Y
Star tracker configuration File	1.0	EO_OPER_INT_STRCFG_0100.XSD	-	N
	1.1	EO_OPER_INT_STRCFG_0101.XSD	-	R
	1.2	EO_OPER_INT_STRCFG_0102.XSD	-	R
	2.0	EO_OPER_INT_STRCFG_0200.XSD	-	R
	2.1	EO_OPER_INT_STRCFG_0201.XSD	-	R
DEM Configuration File	1.0	EO_OPER_INT_DEMCFG_0100.XSD	-	N
	1.1	EO_OPER_INT_DEMCFG_0101.XSD	-	R
	1.2	EO_OPER_INT_DEMCFG_0102.XSD	-	R
	1.3	EO_OPER_INT_DEMCFG_0103.XSD	-	R
	1.4	EO_OPER_INT_DEMCFG_0104.XSD	-	R
	1.5	EO_OPER_INT_DEMCFG_0105.XSD	-	R
	1.6	EO_OPER_INT_DEMCFG_0106.XSD	-	R
	2.0	EO_OPER_INT_DEMCFG_0200.XSD	-	R
	2.1	EO_OPER_INT_DEMCFG_0201.XSD	-	R
	2.2	EO_OPER_INT_DEMCFG_0202.XSD	-	R
	2.3	EO_OPER_INT_DEMCFG_0203.XSD	-	R
Swath Definition File	1.0	EO_OPER_MPL_SW_DEF_0100.XSD	-	N
	1.1	EO_OPER_MPL_SW_DEF_0101.XSD	-	R
	2.1	EO_OPER_MPL_SW_DEF_0201.XSD	-	R
	2.2	EO_OPER_MPL_SW_DEF_0202.XSD	-	R
	3.0	EO_OPER_MPL_SW_DEF_0300.XSD	-	R
	3.1	EO_OPER_MPL_SW_DEF_0301.XSD	-	R

File type	Version	Schema	Mission	Supp. CFI's
Swath Template File	1.0	EO_OPER_MPL_SWTREF_0100.XSD	-	N
	1.1	EO_OPER_MPL_SWTREF_0101.XSD	-	R
	2.0	EO_OPER_MPL_SWTREF_0200.XSD	-	R
	2.1	EO_OPER_MPL_SWTREF_0201.XSD	-	R
	2.2	EO_OPER_MPL_SWTREF_0202.XSD	Sat_1	Y
			Sat_2	N
	3.0	EO_OPER_MPL_SWTREF_0300.XSD	-	R
	3.1	EO_OPER_MPL_SWTREF_0301.XSD	Sat_1	N
			Sat_2	Y
Zone Database File	1.0	EO_OPER_MPL_ZON_DB_0100.XSD	-	N
	1.1	EO_OPER_MPL_ZON_DB_0101.XSD	-	R
	1.2	EO_OPER_MPL_ZON_DB_0102.XSD	-	R
	1.3	EO_OPER_MPL_ZON_DB_0103.XSD	-	R
	2.0	EO_OPER_MPL_ZON_DB_0200.XSD	-	R
	2.1	EO_OPER_MPL_ZON_DB_0201.XSD	-	R
Station Database File	1.0	EO_OPER_MPL_GND_DB_0100.XSD	-	N
	1.1	EO_OPER_MPL_GND_DB_0101.XSD	-	R
	1.2	EO_OPER_MPL_GND_DB_0102.XSD	-	R
	1.3	EO_OPER_MPL_GND_DB_0103.XSD	-	R
	1.4	EO_OPER_MPL_GND_DB_0104.XSD	-	R
	1.5	EO_OPER_MPL_GND_DB_0105.XSD	-	R
	2.0	EO_OPER_MPL_GND_DB_0200.XSD	-	R
	2.1	EO_OPER_MPL_GND_DB_0201.XSD	-	R
Precise Propagator Configuration File	1.0	EO_OPER_INT_PPRCFG_0100.XSD	-	R
	1.1	EO_OPER_INT_PPRCFG_0101.XSD	-	R
	2.0	EO_OPER_INT_PPRCFG_0200.XSD	-	R
	2.1	EO_OPER_INT_PPRCFG_0201.XSD	-	R

File type	Version	Schema	Mission	Supp. CFI's
TLE File	-	TLE file is not in XML	-	-

For consulting the format of these files in their last version see section 9. For older versions, the format can be consulted through the schema.

In the CFI's installation packages schemas and example files are included (under *files* directory, see [GEN_SUM]).

9. FILES FORMAT SPECIFICATION

This section presents the formats for all the files used by the Earth Observation CFI software. These formats correspond to the last schema versions indicated in section 8.

The files used by the CFI can be:

- External: Files generated and/or used for the CFI software and other external facilities.
- Internal: Files used only in the CFI for configuration purposes.

All internal files are written in ASCII, with XML syntax. Following the usual format for the Earth Observation Files, the file contains both:

- A header: It is divided in a fixed header and optionally a variable header. The format for the fixed header is common to all Earth Observation Files and can be seen in [EE_FMT].
- A data block containing the input/output data of the functions.
- The general structure for a file will be:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<Earth_Explorer_File>
   <Earth_Explorer_Header>
      <Fixed_Header>
      ...
      </Fixed_Header>
      <Variable_Header>
      ...
      </Variable_Header>
   </Earth_Explorer_Header>


   <Data_Block type="xml">
   ...
   </Data_Block type>


</Earth_Explorer_File>
```

9.1.Fixed Header

9.1.1.Format

The Fixed Header is an XML structure. Many of its fields are redundant with the File Name elements, but are present in more readable form in the Fixed Header, whereas in File Name they are more compact for obvious reasons. Its format is described in the followig tables:

Table 129: Fixed Header Structure

Tag name	type	Attribute	C Format	Description

File_Name	string	-	%s	It is a repetition of the Logical File Name, i.e. the File Names excluding the extension. This allows this field to be independent from the storage in 1 complete file or 2 separate files for Header and Data Block
File_Description	string	-	%s	A 1-line description of the File Type. Each Mission shall define the list of official file descriptions (per File Type).
Notes	string	-	%s	Multi-lines free text. This can be used for any type of comment, relevant that instance of the file.
Mission	string	-	%s	A 1-word description of the Mission, coherent with the Mission element in the File Name. See [EE_FMT] Section 4.1.1 for the official list.
File_Class	string	-	%10s	A 1-line description of the file class, coherent with the File Class element in the File Name. Each Mission shall define the list of official file classes.
File_Type	string	-	%s	It is a repetition of the File Type element in the File Name.
Validity_Period	structure (see Table 130)	-	-	Structure containing the start-stop validity period of the file.
File_Version	integer	-	%04ld	It is a repetition of the File Version element in the File Name. Must start at 1 (not 0).
Source	structure (see Table 131)	-	-	Structure with information about the source of the file.

Table 130: Fixed Header. Validity Period

Tag name	type	Attribute	C Format	Description
Validity_Start	%23s	-	string	This is the UTC Validity Start Time, coherent with the Validity Start Time in the File Name, but in CCSDS ASCII format with time reference. Note that this can have the special value indicating “beginning of mission” (without an absolute time specified) as defined in [MCD]

Validity_Stop	%23s	-	string	This is the UTC Validity Stop Time, coherent with the Validity Stop Time in the File Name, but in CCSDS ASCII format with time reference. Note that this can have the special value indicating “end of mission” (without an absolute time specified) as defined in [MCD].

Table 131: Fixed Header. Source

Tag name	type	Attribute	C Format	Description
System	string	-	%s	Name of the Ground Segment element creating the file (e.g. FOS, PDS, SSALTO...)
Creator	string	-	%s	Name of the tool, within the Ground Segment element, creating the file (e.g. CS-MCS, IPF1...)
Creator_Version	string	-	%s	Version of the tool (e.g. 1.0, 2.1a ...)
Creation_Date	string	-	%23s	This is the UTC Creation Date, in CCSDS ASCII format with time reference. This format is defined in [MCD].

9.1.2.Example

```
<Fixed_Header>
        <File_Name>logical file name</File_Name>
        <File_Description>1-line file description</File_Description>
        <Notes>
             free text, free format
             several lines if needed

        </Notes>
        <Mission>mission name</Mission> (e.g. Cryosat)
        <File_Class>1-line file class description</File_Class>
        <File_Type>TTTTTTTTTT</File_Type>
        <Validity_Period>
             <Validity_Start>UTC=yyyy-mm-ddThh:mm:ss</Validity_Start>
             <Validity_Stop>UTC=yyyy-mm-ddThh:mm:ss</Validity_Stop>
        </Validity_Period>
        <File_Version>vvvv</File_Version>
        <Source>
             <System>name of system creating the file</System>
             <Creator>name of tool creating the file</Creator>
             <Creator_Version>version of tool</Creator_Version>
             <Creation_Date>UTC=yyyy-mm-ddThh:mm:ss</Creation_Date>
        </Source>
</Fixed_Header>
```

9.2.Predicted Orbit files

9.2.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT]
2. Variable Header: It contains the information for of the reference frame of the state vectors in the file and the reference time for time recomputations. The format is in Table 132.
3. Data Block: It consists in a set of structures described in the tables below:

Table 132: Predicted Orbit File. Variable_Header

Tag name	type	Attribute	C Format	Description
Ref_Frame	string	-	%s	Reference frame for the state vectors in the file. It can be one of the following values: BAR_MEAN_2000 HEL_MEAN_2000 GEO_MEAN_2000 MEAN_DATE TRUE_DATE EARTH_FIXED BAR_MEAN_1950 QUASI_MEAN_DATE PSE_TRUE_DATE PSEUDO_EARTH_FIXED
Time_Reference	String	-	%s	It is used as time reference when, due to an inconsistency with other time correlations, times associated to the state vector have to be recomputed. See xo_orbit_init_file in [ORBIT_SUM] for more details. It can be one of the following values:TAIUTC UT1

Note to Reference Frame: the main usage of the Orbit Library is to support geo-location. As a consequence an Earth-Fixed frame is the natural reference frame to use, end-to-end. On the other hand, accurate conversion between inertial and earth-fixed frames happens only when polar motion data is available, more precisely when the time_id has been previously initialized with a IERS Bulletin.

Therefore, when polar motion data is not available, the user is recommended to initialize the orbit id by providing Orbit files with Earth-Fixed frame data (orbit state vectors), and to compute geo-location information in Earth-Fixed. This provides accurate computations. For any other usage the user shall be aware of the consequences and accept small inaccuracies, In particular:

•initializing the orbit id with inertial orbit data, and computing inertial parameters, is also supported and is accurate;

•initializing the orbit id with earth-fixed orbit data, and computing inertial parameters (or vice-cersa), leads to slightly inaccurate computations and should be avoided unless ignoring polar motion is acceptable.

Table 133: Predicted Orbit File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_OSVs	List of <OSV> Structures (See Table 134)	count="*n*" where *n* is the number of elements in the list	-	List of Orbit State Vectors

Table 134: Precicted Orbit File. OSV

Tag name	type	Attribute	C Format	Description
TAI	date		%s	TAI date and time of OSV, in ASCII standard time format, including time reference and micro-seconds
UTC	date		%s	UTC date and time of OSV, in ASCII standard time format, including time reference and micro-seconds
UT1	date		%s	UT1 date and time of OSV, in ASCII standard time format, including time reference and micro-seconds
Absolute_Orbit	integer		%+06ld	absolute orbit counter This counter is incremented by one unit from a record to the next. It must be dif- ferenciated with the real absolute orbit number on which the state vector really belongs i.e : if the Z value of the OSV is >= 0 then “real” absolute orbit number equal the absolute orbit counter if the Z value of the OSV is < 0 then “real” absolute orbit number equal the absolute orbit counter minus 1.
X	real	m	%+012.3lf	X position in earth-fixed coordinate system
Y	real	m	%+012.3lf	Y position in earth-fixed coordinate system
Z	real	m	%+012.3lf	Z position in earth-fixed coordinate system
VX	real	m/s	%+012.6lf	X velocity in earth-fixed coordinate system

VY	real	m/s	%+012.6lf	Y velocity in earth-fixed coordinate sys-tem
VZ	real	m/s	%+012.6lf	Z velocity in earth-fixed coordinate system
Quality	string	string	%13s	Values is/are TBD. This parameter is added to keep format compatibility with the DORIS Precise Orbit File Format. Default (“not used”) value is “0000000000000”

9.2.2.Example

```
<?xml version ="1.0"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
 EO_OPER_MPL_ORBPRE_01032.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.32">
       <Earth_Explorer_Header>
             <Fixed_Header>
             <File_Name>CS_OPER_MPL_ORBPRE_20020315T205400_20020321T205500_00032</File_Name>
             <File_Description>FOS Predicted Orbit File</File_Description>
             <Notes></Notes>
             <Mission>CryoSat</Mission>
             <File_Class>Routine Operations</File_Class>
             <File_Type>MPL_ORBPRE</File_Type>
             <Validity_Period>
                   <Validity_Start>UTC=2002-03-15T20:54:44</Validity_Start>
                   <Validity_Stop>UTC=2002-03-21T20:54:44</Validity_Stop>
             </Validity_Period>
             <File_Version>0001</File_Version>
             <Source>
                   <System>FOS</System>
                   <Creator>name of tool creating the file</Creator>
                   <Creator_Version>1.0</Creator_Version>
                   <Creation_Date>UTC=2002-03-14T14:00:00</Creation_Date>
             </Source>
             </Fixed_Header>
             <Variable_Header>
                   <Ref_Frame>EARTH_FIXED</Ref_Frame>
                   <Time_Reference>UTC</Time_Reference>
             </Variable_Header>
       </Earth_Explorer_Header>
       <Data_Block type=”xml”>
             <List_of_OSVs count=”n”>
                   <OSV>
                         <TAI>TAI=2002-03-15T20:54:44.069916</TAI>
                         <UTC>UTC=2002-03-15T20:54:04.069916</UTC>
                         <UT1>UT1=2002-03-15T20:54:04.049916</UT1>
                         <Absolute_Orbit>+00212</Absolute_Orbit>
                         <X unit=”m”>+6874869.308</X>
                         <Y unit=”m”>+2033241.443</Y>
                         <Z unit="m">-0000995.334</Z>
                         <VX unit="m/s">+0453.224305</VX>
                         <VY unit="m/s">-1567.965124</VY>
                         <VZ unit="m/s">+7374.880929</VZ>
                         <Quality>0000000000000</Quality>
                   </OSV>
                   <OSV>
                         <TAI>TAI=2002-03-15T22:35:24.246686</TAI>
                         <UTC>UTC=2002-03-15T22:34:44.246686</UTC>
                         <UT1>UT1=2002-03-15T22:34:44.046686</UT1>
                         <Absolute_Orbit>+00213</Absolute_Orbit>
                         <X unit="m">+7086938.577</X>
                         <Y unit="m">-1083333.239</Y>
                         <Z unit="m">-0001004.069</Z>
                         <VX unit="m/s">-0256.608063</VX>
                         <VY unit="m/s">-1611.943172</VY>
                         <VZ unit="m/s">+7374.846086</VZ>
                         <Quality>0000000000000</Quality>
                   </OSV>
                   ...
               </List_of_OSVs>
        </Data_Block>
</Earth_Explorer_File>
```

9.3.Restituted Orbit files

Format is equal to the Predicted Orbit File, see section 9.2.

9.4.Doris Preliminary/Precise files

Format is equal to the Predicted Orbit File, see section 9.2.

9.5.Orbit Scenario files

9.5.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT]
2. Variable Header: EmptyIt contains the information of the reference time for time recomputations. The format is in Table 135.
3. Data Block: It consists in a set of structures described in the tables below:

Table 135: Orbit Scenario File. Variable_Header

Tag name	type	Attribute	C Format	Description

Time_Reference	String	-	%s	Reference time. For Orbit Scenario Files only the following value is allowed: UT1

Table 136: Orbit Scenario File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_Orbit_Changes	List of <Orbit_Change> Structures (See Table 137)	count=”*n*” where *n* is the number of elements in the list	-	List of Orbital Changes

Table 137: Orbit Scenario File. Orbit_Change

Tag name	type	Attribute	C Format	Description
Orbit	structure (see Table 138)			Orbit information
Cycle	structure (see Table 139)			Cycle information
Time_of_ANX	structure (see Table 142)			Ascending node time

Table 138: Orbit Scenario File. Orbit

Tag name	type	Attribute	C Format	Description
Absolute_Orbit	integer		%+06ld	absolute orbit counter.
Relative_Orbit	integer		%ld	relative orbit number
Cycle_Number	integer		%ld	cycle number; incremented after each new repeat cycle
Phase_Number	integer		%ld	phase number; incremented on Mission Management decision

Table 139: Orbit Scenario File. Cycle

Tag name	type	Attribute	C Format	Description
Repeat_Cycle	integer		%ld	
Cycle_Length	integer		%ld	
ANX_Longitude	real	deg	%+011.6lf	longitude of ascending node crossing

MLST	date		%s	mean local solar time at ANX of relative orbit 1
MLST_Drift	real	s/day	%+12.6lf	drift of mean local solar time over 1 orbit
MLST_Nonlinear_Drift	structure(see Table 140)			Non linear MLST data

Table 140: MLST non linear terms

Tag name	type	Attribute	C Format	Description
Linear_Approx_Validity	integer	orbits	20 (fixed value)	Number of orbits in which linear approximation is valid.
Quadratic_Term	real	s/day^2	%8.6lf	MLST Quadratic term.
Harmonics_Terms	List of <Harmonic_Term> structures (see Table 141)	num=”num_harm” where num_harm is the number of harmonics (between 0 and 2)		List of harmonic terms.

Table 141: MLST Harmonic term

Tag name	type	Attribute	C Format	Description
Reference_Time	date	time_ref=”UT1”	%s	Reference time of harmonic
Period	real	unit=”days”	%6.2lf	Period
Amplitude_Sin	real	unit=”sec”	%5.3lf	Amplitude of sine
Amplitude_Cos	real	unit=”sec”	%5.3lf	Amplitude of cosine

Table 142 Orbit Scenario File. Time_of_ANX

Tag name	type	Attribute	C Format	Description
TAI	date		%s	TAI date and time of ANX, in ASCII CCSDStime format, including time ref- erence and micro-seconds
UTC	date		%s	UTC date and time of ANX, in ASCII CCSDS time format, including time ref- erence and micro-seconds
UT1	date		%s	UT1 date and time of ANX, in ASCII CCSDS time format, including time ref- erence and micro-seconds

9.5.2.Example

```
<?xml version ="1.0"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
 EO_OPER_MPL_ORBSCT_01021.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.12">
        <Earth_Explorer_Header>
              <Fixed_Header>
              <File_Name>CS_OPER_MPL_ORBSCT_20020312T140002_99999999T999999_00021</File_Name>
              <File_Description>Orbit Scenario File</File_Description>
              <Notes></Notes>
              <Mission>CryoSat</Mission>
              <File_Class>Routine Operations</File_Class>
              <File_Type>MPL_ORBSCT</File_Type>
              <Validity_Period>
                   <Validity_Start>UTC=2002-03-12T14:00:02</Validity_Start>
                   <Validity_Stop>UTC=9999-99-99T99:99:99</Validity_Stop>
              </Validity_Period>
              <File_Version>0001</File_Version>
              <Source>
                   <System>RPF</System>
                   <Creator>name of tool creating the file</Creator>
                   <Creator_Version>1.0</Creator_Version>
                   <Creation_Date>UTC=2002-03-10T14:00:00</Creation_Date>
              </Source>
              </Fixed_Header>
              <Variable_Header>
                   <Time_Reference>UT1</Time_Reference>
              </Variable_Header>
         </Earth_Explorer_Header>
         <Data_Block type="xml">
              <List_of_Orbit_Changes count="2">
                   <Orbit_Change>
                          <Orbit>
                               <Absolute_Orbit>+00001</Absolute_Orbit>
                              <Relative_Orbit>1</Relative_Orbit>
                              <Cycle_Number>1</Cycle_Number>
                              <Phase_Number>1</Phase_Number>
                          </Orbit>
                          <Cycle>
                              <Repeat_Cycle unit="day">369</Repeat_Cycle>
                              <Cycle_Length unit="orbit">5344</Cycle_Length>
                              <ANX_Longitude unit="deg">+023.600000</ANX_Longitude>
                              <MLST>22:17:19.999999</MLST>
                              <MLST_Drift unit="s/day">+9.000000</MLST_Drift>
                              <MLST_NonlinearDrift>
                               <Linear_Approx_Validity unit=”orbits”>20</Linear_Approx_Validity>
                               <Quadratic_Term unit=”s/day^2”>2.345678</Quadratic_Term>
                               <Harmonics_Terms num=”2”>
                                <Harmonic_Term seq=”1”>
                                 <Reference_Time time_ref=”UT1”>2001-03-13T00:00:00.000000
                                 </Reference_Time>
                                 <Period unit=”days”>100.00</Period>
                                 <Amplitude_Sin unit=”sec”>2.222</Amplitude_Sin>
                                 <Amplitude_Cos unit=”sec”>1.111</Amplitude_Cos>
                                </Harmonic_Term>
                                <Harmonic_Term seq=”2”>
```

```
                                  <Reference_Time time_ref=”UT1”>2001-03-14T00:00:00.000000
                                  </Reference_Time>
                                  <Period unit=”days”>100.00</Period>
                                  <Amplitude_Sin unit=”sec”>3.333</Amplitude_Sin>
                                  <Amplitude_Cos unit=”sec”>4.444</Amplitude_Cos>
                                 </Harmonic_Term>
                                </Harmonics_Terms>
                               </MLST_NonlinearDrift>
                         </Cycle>
                         <Time_of_ANX>
                               <TAI>TAI=2001-03-12T14:00:34.999999</TAI>
                               <UTC>UTC=2001-03-12T14:00:02.999999</UTC>
                               <UT1>UT1=2001-03-12T14:00:02.777777</UT1>
                         </Time_of_ANX>
                   </Orbit_Change>
                   <Orbit_Change>
                         <Orbit>
                               <Absolute_Orbit>+00050</Absolute_Orbit>
                               <Relative_Orbit>1</Relative_Orbit>
                               <Cycle_Number>10</Cycle_Number>
                               <Phase_Number>1</Phase_Number>
                         </Orbit>
                         <Cycle>
                               <Repeat_Cycle unit="day">2</Repeat_Cycle>
                               <Cycle_Length unit="orbit">29</Cycle_Length>
                               <ANX_Longitude unit="deg">+023.600000</ANX_Longitude>
                               <MLST>22:17:19.999999</MLST>
                               <MLST_Drift unit="s/day">+9.000000</MLST_Drift>
                               <MLST_NonlinearDrift>
                                <Linear_Approx_Validity unit=”orbits”>20</Linear_Approx_Validity>
                                <Quadratic_Term unit=”s/day^2”>2.345678</Quadratic_Term>
                                <Harmonics_Terms num=”2”>
                                 <Harmonic_Term seq=”1”>
                                  <Reference_Time time_ref=”UT1”>2001-03-13T00:00:00.000000
                                  </Reference_Time>
                                  <Period unit=”days”>100.00</Period>
                                  <Amplitude_Sin unit=”sec”>2.222</Amplitude_Sin>
                                  <Amplitude_Cos unit=”sec”>1.111</Amplitude_Cos>
                                 </Harmonic_Term>
                                 <Harmonic_Term seq=”2”>
                                  <Reference_Time time_ref=”UT1”>2001-03-14T00:00:00.000000
                                  </Reference_Time>
                                  <Period unit=”days”>100.00</Period>
                                  <Amplitude_Sin unit=”sec”>3.333</Amplitude_Sin>
                                  <Amplitude_Cos unit=”sec”>4.444</Amplitude_Cos>
                                 </Harmonic_Term>
                                </Harmonics_Terms>
                               </MLST_NonlinearDrift>
                         </Cycle>
                         <Time_of_ANX>
                               <TAI>TAI=2001-04-01T14:00:34.999999</TAI>
                               <UTC>UTC=2001-04-01T14:00:02.999999</UTC>
                               <UT1>UT1=2001-04-01T14:00:02.777777</UT1>
                         </Time_of_ANX>
                   </Orbit_Change>
              </List_of_Orbit_Changes>
         </Data_Block>
</Earth_Explorer_File>
```

9.6.Orbit Event files

Orbit Event File is deprecated and is supported only for Cryosat mission.

9.6.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: It contains the information for of the reference frame of the state vectors in the file and the reference time for time recomputations. The format is in Table 132.
3. Data Block: It consists in a set of structures described in the tables below:

Table 143: Orbit Scenario File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_Orbit_Changes	List of <Orbit_Change> Structures (See Table 137)	count=”n” where n is the number of elements in the list	-	List of Orbital Changes
List_of_OSVs	List of <OSV> Structures (See Table 134)	count=”n” where n is the number of elements in the list	-	List of Orbit State Vectors

9.6.2.Example

```
  <?xml version="1.0"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”

 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
 EO_OPER_MPL_ORBREF_01032.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.32">

   <Earth_Explorer_Header>
     <Fixed_Header>
      <File_Name>CS_TEST_MPL_ORBREF_20020303T080916_20020303T094823_00031</File_Name>
      <File_Description>Orbit Event File</File_Description>
      <Notes/>
      <Mission>CryoSat</Mission>
      <File_Class>TEST</File_Class>
      <File_Type>MPL_ORBREF</File_Type>
      <Validity_Period>
        <Validity_Start>UTC=2002-03-03T08:09:16</Validity_Start>
        <Validity_Stop>UTC=2002-03-03T09:48:23</Validity_Stop>
      </Validity_Period>
      <File_Version>0101</File_Version>
      <Source>
        <System>CFI Acceptance</System>
        <Creator>EO_ORBIT: xo_gen_oef</Creator>
        <Creator_Version>3.5</Creator_Version>
     <Creation_Date>UTC=2006-09-18T13:21:11</Creation_Date>
   </Source>
  </Fixed_Header>
  <Variable_Header>
   <Ref_Frame>EARTH_FIXED</Ref_Frame>
   <Time_Reference>UTC</Time_Reference>
  </Variable_Header>
</Earth_Explorer_Header>
<Data_Block type="xml">
   <List_of_Orbit_Changes count="2">
   <Orbit_Change>
     <Orbit>
       <Absolute_Orbit>1</Absolute_Orbit>
       <Relative_Orbit>25</Relative_Orbit>
       <Cycle_Number>1</Cycle_Number>
       <Phase_Number>1</Phase_Number>
     </Orbit>
     <Cycle>
       <Repeat_Cycle unit="day">2</Repeat_Cycle>
       <Cycle_Length unit="orbit">29</Cycle_Length>
       <ANX_Longitude unit="deg">130.000000</ANX_Longitude>
       <MLST>21:00:00.000000</MLST>
       <MLST_Drift unit="s/day">-179.045927</MLST_Drift>
     </Cycle>
     <Time_of_ANX>
       <TAI>TAI=2002-03-01T21:00:52.365827</TAI>
       <UTC>UTC=2002-03-01T21:01:27.365827</UTC>
       <UT1>UT1=2002-03-01T21:01:27.665827</UT1>
     </Time_of_ANX>
   </Orbit_Change>
   <Orbit_Change>
     <Orbit>
       <Absolute_Orbit>30</Absolute_Orbit>
       <Relative_Orbit>1864</Relative_Orbit>
       <Cycle_Number>2</Cycle_Number>
       <Phase_Number>1</Phase_Number>
     </Orbit>
     <Cycle>
       <Repeat_Cycle unit="day">369</Repeat_Cycle>
       <Cycle_Length unit="orbit">5344</Cycle_Length>
       <ANX_Longitude unit="deg">129.998600</ANX_Longitude>
       <MLST>20:54:02.999999</MLST>
       <MLST_Drift unit="s/day">-179.208551</MLST_Drift>
     </Cycle>
     <Time_of_ANX>
       <TAI>TAI=2002-03-03T20:46:50.497469</TAI>
       <UTC>UTC=2002-03-03T20:47:25.497469</UTC>
       <UT1>UT1=2002-03-03T20:47:25.797469</UT1>
     </Time_of_ANX>
   </Orbit_Change>
 </List_of_Orbit_Changes>
 <List_of_OSVs count="2">
   <OSV>
     <TAI>TAI=2002-03-03T08:08:41.244734</TAI>
     <UTC>UTC=2002-03-03T08:09:16.244734</UTC>
     <UT1>UT1=2002-03-03T08:09:16.544734</UT1>
```

```
       <Absolute_Orbit>+00013</Absolute_Orbit>
       <X unit="m">-6937171.769</X>
       <Y unit="m">-1483270.979</Y>
       <Z unit="m">+0000000.000</Z>
       <VX unit="m/s">-0152.952889</VX>
       <VY unit="m/s">+0761.962112</VY>
       <VZ unit="m/s">+7493.050200</VZ>
       <Quality>000000.000000</Quality>
     </OSV>
     <OSV>
       <TAI>TAI=2002-03-03T09:47:47.517429</TAI>
       <UTC>UTC=2002-03-03T09:48:22.517429</UTC>
       <UT1>UT1=2002-03-03T09:48:22.817429</UT1>
       <Absolute_Orbit>+00014</Absolute_Orbit>
       <X unit="m">-6918815.899</X>
       <Y unit="m">+1566662.540</Y>
       <Z unit="m">+0000000.000</Z>
       <VX unit="m/s">+0181.123304</VX>
       <VY unit="m/s">+0755.761334</VY>
       <VZ unit="m/s">+7493.050200</VZ>
       <Quality>000000.000000</Quality>
     </OSV>
    </List_of_OSVs>
  </Data_Block>
</Earth_Explorer_File>
```

9.7.DORIS Navigator files

A DORIS Navigator file consist in two files, the header file and the data block file. They are compliant with [PDS_FMT]

9.8.Star Tracker files

A Star tracker file consists in a couple of files: the CryoSat standard header file and the data block file. They are compliant with [PDS_FMT]

9.9.Satellite Configuration File

9.9.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 144: Satellite Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
Satellite_Name	string	-	%s	Satellite Name
NORAD_Data	structure (see Table 146)	-	-	NORAD Satellite data
Lib_Init	structure (see Table 146)	-	-	Low and tight tolerances for orbital parameters
Orbit_Init	)	-	-	Default Orbital parameters

Table 145: Satellite Configuration File. NORAD_Data Structure

Tag name	type	Attribute	C Format	Description
Satellite_Number	integer	-	%ld	NORAD Satellite number
NORAD_Sat_Name	string	-	%s	NORAD Satellite name
Int_Designator	string	-	%s	NORAD international designator

Table 146: Satellite Configuration File. Lib_Init Structure

Tag name	type	Attribute	C Format	Description
Low_Tolerances	structure (see Table 148)	-	-	Low tolerances for orbital parameters
Tight_Tolerances	structure (see Table 144)	-	-	Tight tolerances for orbital parameters

Table 147: Satellite Configuration File. Orbit_InitStructure

Tag name	type	Attribute	C Format	Description
Min_Semi_Major_Axis	real	-	%lf	Minimum semi-major axis (meters)
Nom_Semi_Major_Axis	real	-	%lf	Nominal semi-major axis (meters)

Max_Semi_Major_Axis	real	-	%lf	Maximum semi-major axis (meters)
Min_Inclination	real	-	%lf	Minimum inclination (degrees)
Nom_Inclination	real	-	%lf	Nominal inclination (degrees)
Max_Inclination	real	-	%lf	Maximum inclination (degrees)
Nom_Eccentricity	real	-	%lf	Nominal Eccentricity
Nom_Arg_Perigee	real	-	%lf	Nominal Argument of perigee (degrees)

Table 148: Satellite Configuration File. Low and Tight Tolerances Structure

Tag name	type	Attribute	C Format	Description
Min_Semi_Major_Axis	real	-	%lf	Minimum semi-major axis (meters)
Max_Semi_Major_Axis	real	-	%lf	Maximum semi-major axis (meters)
Min_Inclination	real	-	%lf	Minimum inclination (degrees)
Max_Inclination	real	-	%lf	Maximum inclination (degrees)
Min_Eccentricity	real	-	%lf	Eccentricity
Max_Eccentricity	real	-	%lf	Eccentricity

9.9.2.File Example

```
 <?xml version = "1.0" encoding = "UTF-8"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
 EO_OPER_INT_SATCFG_0102.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.2">

    <Earth_Explorer_Header>
      <Fixed_Header>
        <File_Name>Cryosat_configuration_file.xml</File_Name>
        <File_Description>Satellite Configurantion File</File_Description>
        <Notes/>
        <Mission>Cryosat</Mission>
        <File_Class>TEST</File_Class>
        <File_Type></File_Type>
        <Validity_Period>
            <Validity_Start>UTC=0000-00-00T00:00:00.000000</Validity_Start>
            <Validity_Stop>UTC=9999-99-99T99:99:99.5999999</Validity_Stop>
      </Validity_Period>
      <File_Version>1</File_Version>
      <Source>
        <System></System>
        <Creator></Creator>
        <Creator_Version></Creator_Version>
        <Creation_Date>UTC=2003-11-28T17:25:44</Creation_Date>
      </Source>
      </Fixed_Header>
      <Variable_Header\>
    </Earth_Explorer_Header>
    <Data_Block type="xml">
        <Satellite_Name>CryoSat</Satellite_Name>
        <NORAD_Data>
               <Satellite_Number>00000</Satellite_Number>
              <NORAD_Sat_Name>CRYOSAT</NORAD_Sat_Name>
              <Int_Designator>00000</Int_Designator>
        </NORAD_Data>
        <Lib_Init>
              <Low_Tolerances>
                    <Min_Semi_Major_Axis>1000000.0</Min_Semi_Major_Axis>
                    <Max_Semi_Major_Axis>10000000.0</Max_Semi_Major_Axis>
                    <Min_Inclination>60.0</Min_Inclination>
                    <Max_Inclination>120.0</Max_Inclination>
                    <Min_Eccentricity>0.0</Min_Eccentricity>
                    <Max_Eccentricity>0.5</Max_Eccentricity>
              </Low_Tolerances>
              <Tight_Tolerances>
                    <Min_Semi_Major_Axis>1000000.0</Min_Semi_Major_Axis>
                    <Max_Semi_Major_Axis>10000000.0</Max_Semi_Major_Axis>
                    <Min_Inclination>60.0000</Min_Inclination>
                    <Max_Inclination>120.0000</Max_Inclination>
                    <Min_Eccentricity>0.000</Min_Eccentricity>
                    <Max_Eccentricity>0.500</Max_Eccentricity>
              </Tight_Tolerances>
        </Lib_Init>
        <Orbit_Init>
              <Min_Semi_Major_Axis>7055200.0</Min_Semi_Major_Axis>
              <Nom_Semi_Major_Axis>7096643.0</Nom_Semi_Major_Axis>
              <Max_Semi_Major_Axis>7131206.0</Max_Semi_Major_Axis>
              <Min_Inclination>91.8981</Min_Inclination>
              <Nom_Inclination>92.0000</Nom_Inclination>
              <Max_Inclination>92.0732</Max_Inclination>
              <Nom_Eccentricity>0.0013</Nom_Eccentricity>
              <Nom_Arg_Perigee>90.0</Nom_Arg_Perigee>
        </Orbit_Init>
</Data_Block>
</Earth_Explorer_File>
```

9.10.Attitude File

9.10.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 149: Attitude File. Data Block

Tag name	type	Attribute	C Format	Description
Attitude_File_Type	string	-	%s	The initial attitude frame. It can be: • Sat_Nominal_Attitude • Sat_Attitude • Instr_Attitude
Attitude_Data_Type	string	-	%s	It defines the type of attitude data: • Quaternions • Attitude_Angles
Max_Gap		unit="s"		Maximum gap between two consecutive set of angles or quaternions
Attitude_Angles_Data or Quaternions_Data	Structures: see Table 150 for the angles data or Table 151 for the quaternions	-	-	Structure for the list of angles or the quaternions

Table 150: Attitude File. Attitude Angles Data

Tag name	type	Attribute	C Format	Description
List_of_Attitude_Angles	List of Attitude_Angles (see Table 152)	count="n"	-	List of Attitude_Angles

Table 151: Attitude File. Quaternions Data

Tag name	type	Attribute	C Format	Description
Inertial_Ref_Frame	string	-	%s	Inertial reference frame. It could be one of the following: BM2000 HM2000 GM2000 MEAN_DATE TRUE_DATE QUASI_MEAN_DATE PSEUDO_TRUE_DATE
List_of_Quaternions	List of Quaternions (see Table 153)	count=”n”	-	List of Quaternions

Table 152: Attitude File. List of Attitude Angles

Tag name	type	Attribute	C Format	Description
Attitude_Angles	Structure (see Table 154)	-	-	Pitch, roll and yaw angles for a given time. When angles are used to create an attitude file, 6 decimal digits are written (for quaternions 9 decimal digits are written, see Table 153).

Table 153: Attitude File. List of Quaternions Data

Tag name	type	Attribute	C Format	Description
Quaternions	Structure (see Table 155)	-	-	Set of quaternions for a given time. When quaternions are used to create an attitude file, 9 decimal digits are written. Having this number of digits reduces significantly pointing errors.

Table 154: Attitude File. Attitude_Angles

Tag name	type	Attribute	C Format	Description
Time	string	ref=”RRR” where RRR stands for: • TAI • UTC • UT1 • GPS	%s	Date for the angles. The date format is CCSDS-A with reference and microseconds (RRR=yyyy-mm-ddThh:nn:ss.uuuuuu)
Pitch	real	unit=”deg”	%lf	Pitch angle

Roll	real	unit=”deg”	%lf	Roll angle
Yaw	real	unit=”deg”	%lf	Yaw angle

Table 155: Attitude File. Quaternions

Tag name	type	Attribute	C Format	Description
Time	string	ref=”RRR” where RRR stands for: • TAI • UTC • UT1 • GPS	%s	Date for the angles. The date format is CCSDS-A with reference and microseconds (RRR=yyyy-mm-ddThh:nn:ss.uuuuuu)
Q1	real	-	%lf	Quaternion
Q2	real	-	%lf	Quaternion
Q3	real	-	%lf	Quaternion
Q4	real	-	%lf	Quaternion

9.10.2.File Example with angles

```
<?xml version="1.0"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_INT_ATTREF_0101.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.1">
    <Earth_Explorer_Header>
      <Fixed_Header>
         <File_Name>ATT_TEST_FILE</File_Name>
         <File_Description>Attitude File</File_Description>
         <Notes/>
         <Mission>XXXXX</Mission>
         <File_Class>TEST</File_Class>
         <File_Type></File_Type>
         <Validity_Period>
            <Validity_Start>UTC=2002-03-03T08:09:17.232850</Validity_Start>
            <Validity_Stop>UTC=2002-03-03T09:48:23.505544</Validity_Stop>
         </Validity_Period>
         <File_Version>0101</File_Version>
         <Source>
            <System>CFI Acceptance</System>
            <Creator></Creator>
            <Creator_Version></Creator_Version>
            <Creation_Date>UTC=2003-11-28T17:25:44</Creation_Date>
         </Source>
      </Fixed_Header>
      <Variable_Header/>
    </Earth_Explorer_Header>
    <Data_Block type="xml">
      <Attitude_File_Type>Sat_Attitude</Attitude_File_Type>
      <Attitude_Data_Type>Attitude_Angles</Attitude_Data_Type>
      <Max_Gap unit="s">200</Max_Gap>
      <Attitude_Angles_Data>
         <List_of_Attitude_Angles count="5">
            <Attitude_Angles>
               <Time ref="TAI">TAI=2004-07-04T18:26:30.000000</Time>
               <Pitch unit="deg">0.05</Pitch>
               <Roll unit="deg">0.15</Roll>
               <Yaw unit="deg">-0.25</Yaw>
            </Attitude_Angles>
            <Attitude_Angles>
               <Time ref="TAI">TAI=2004-07-04T18:26:32.000000</Time>
               <Pitch unit="deg">0.07</Pitch>
               <Roll unit="deg">0.17</Roll>
               <Yaw unit="deg">-0.27</Yaw>
            </Attitude_Angles>
            <Attitude_Angles>
               <Time ref="TAI">TAI=2004-07-04T18:26:34.000000</Time>
               <Pitch unit="deg">0.09</Pitch>
               <Roll unit="deg">0.19</Roll>
               <Yaw unit="deg">-0.29</Yaw>
            </Attitude_Angles>
            <Attitude_Angles>
               <Time ref="TAI">TAI=2004-07-04T18:26:36.000000</Time>
               <Pitch unit="deg">0.11</Pitch>
               <Roll unit="deg">0.21</Roll>
               <Yaw unit="deg">-0.31</Yaw>
            </Attitude_Angles>
            <Attitude_Angles>
               <Time ref="TAI">TAI=2004-07-04T18:26:38.000000</Time>
               <Pitch unit="deg">0.13</Pitch>
               <Roll unit="deg">0.23</Roll>
               <Yaw unit="deg">-0.33</Yaw>
            </Attitude_Angles>
            <Attitude_Angles>
               <Time ref="TAI">TAI=2004-07-04T18:26:40.000000</Time>
               <Pitch unit="deg">0.15</Pitch>
               <Roll unit="deg">0.25</Roll>
               <Yaw unit="deg">-0.35</Yaw>
            </Attitude_Angles>
         </List_of_Attitude_Angles>
      </Attitude_Angles_Data>
   </Data_Block>
</Earth_Explorer_File>
```

9.10.3.File Example with quaternions

```
<?xml version="1.0"?>
<Earth_Explorer_File
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://eop-cfi.esa.int/CFI
 http://eop-
 cfi.esa.int/CFI/EE_CFI_SCHEMAS/EO_OPER_INT_ATTREF_0102.
 XSD"
 xmlns="http://eop-cfi.esa.int/CFI" schemaVersion="1.2">
   <Earth_Explorer_Header>
     <Fixed_Header>
      <File_Name>CS_TEST_INT_ATTREF_20020303T080917_20020303T094823_0003</File_Name>
      <File_Description>Attitude File</File_Description>
      <Notes/>
      <Mission>CryoSat</Mission>
   <File_Class>TEST</File_Class>
   <File_Type>INT_ATTCFG</File_Type>
   <Validity_Period>
     <Validity_Start>UTC=2002-03-03T08:09:17</Validity_Start>
     <Validity_Stop>UTC=2002-03-03T09:48:23</Validity_Stop>
   </Validity_Period>
   <File_Version>0102</File_Version>
   <Source>
     <System>CFI Acceptance</System>
     <Creator/>
     <Creator_Version/>
     <Creation_Date>UTC=2003-11-28T17:25:44</Creation_Date>
   </Source>
  </Fixed_Header>
  <Variable_Header/>
</Earth_Explorer_Header>
<Data_Block type="xml">
  <Attitude_File_Type>Sat_Attitude</Attitude_File_Type>
  <Attitude_Data_Type>Quaternions</Attitude_Data_Type>
  <Max_Gap unit="s">200</Max_Gap>
  <Quaternion_Data>
    <Inertial_Ref_Frame>TRUE_DATE</Inertial_Ref_Frame>
    <List_of_Quaternions count="5">
      <Quaternions>
        <Time ref="TAI">TAI=2004-07-04T18:26:33.000000</Time>
        <Q1>0.205523</Q1>
        <Q2>0.694593</Q2>
        <Q3>-0.029401</Q3>
        <Q4>0.688793</Q4>
      </Quaternions>
      <Quaternions>
        <Time ref="TAI">TAI=2004-07-04T18:26:33.600000</Time>
        <Q1>0.205315</Q1>
        <Q2>0.694556</Q2>
        <Q3>-0.029188</Q3>
        <Q4>0.688902</Q4>
      </Quaternions>
      <Quaternions>
        <Time ref="TAI">TAI=2004-07-04T18:26:34.200000</Time>
        <Q1>0.205107</Q1>
        <Q2>0.694518</Q2>
        <Q3>-0.028975</Q3>
        <Q4>0.689011</Q4>
      </Quaternions>
      <Quaternions>
        <Time ref="TAI">TAI=2004-07-04T18:26:34.800000</Time>
        <Q1>0.204899</Q1>
        <Q2>0.694481</Q2>
        <Q3>-0.028762</Q3>
        <Q4>0.689119</Q4>
      </Quaternions>
      <Quaternions>
        <Time ref="TAI">TAI=2004-07-04T18:26:35.400000</Time>
        <Q1>0.204691</Q1>
        <Q2>0.694443</Q2>
```

```
           <Q3>-0.028549</Q3>
           <Q4>0.689228</Q4>
         </Quaternions>
       </List_of_Quaternions>
     </Quaternion_Data>
   </Data_Block>
</Earth_Explorer_File>
```

9.11.Star tracker configuration File

9.11.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below. As it is a quite long file, only the relevant part to the CFIs are described.

Table 156: Star Tracker Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
Satellite_Name	string	-	%s	Satellite Name
Mispointing	Structure (See Table 157)	-	-	Set of rotation angles needed for mispointing computation

Table 157: Star Tracker Configuration File. Mispointing

Tag name	type	Attribute	C Format	Description
Aberration_Correction	string	-	%s	Aberration correction flag. Possible values are: • Yes: for applying the aberration-correction. • No: for not applying the aberration correction. • Reverse: for applying the aberra- tion correction with the transposed matrix.
Star_Trackers_Limits	Structure (See Table 158)	-	-	Limits for the validity fo the quaternions
Star_Trackers_Priority	Structure (See Table 159)	-	-	Star trackers priority
List_of_Star_Trackers	Structure (See Table 160)	count=”n”	-	List of rotation angles from the antenna bench to the star trackers frame
Satellite_Mechanical_To_Antenna_Bench	Structure (See Table 161)	-	-	Rotation angles from the satellite mechanical to the antenna bench frame
Satellite_Control_To_Satellite_Mechanical	Structure (See Table 162)	-	-	Rotation angles from the satellite control to the satellite mechanical frame
Satellite_Attitude_To_Satellite_Control	Structure (See Table 157)	-	-	Rotation angles from the satellite control to the satellite attitude frame

Table 158: Star Tracker Configuration File. Star tracker limits

Tag name	type	Attribute	C Format	Description
Max_Penalty	integer	-	%d	Maximum penalty for the quaternions
Quaternion_Norm_Threshold	real	-	%f	Threshold for the modulus of the quaternion
Max_Time_Gap	real	unit="s"	%f	Maximum time gap between two consequtive quaternions

Table 159: Star Tracker Configuration File. Star_Trackers_Priority

Tag name	type	Attribute	C Format	Description
File_Type_1	string	-	%s	
File_Type_2	string	-	%s	
File_Type_3	string	-	%s	

Table 160: Star Tracker Configuration File. List_of_Star_Trackers

Tag name	type	Attribute	C Format	Description
Star_Tracker	Structure (See Table 161)	-	-	Antenna bench to Star tracker rotation angles

Table 161: Star Tracker Configuration File. Pre and Post Launch angles

Tag name	type	Attribute	C Format	Description
Pre_Launch_Angles	Structure (See Table 162)	-	-	pre-launch angles
Post_Launch_Misalignment	Structure (See Table 162)	-	-	post-launch angles

Table 162: Star Tracker Configuration File. Rotation_Angles

Tag name	type	Attribute	C Format	Description
X_Rotation	real	unit="deg"	%f	Rotation around the X-axis
Y_Rotation	real	unit="deg"	%f	Rotation around the Y-axis
Z_Rotation	real	unit="deg"	%f	Rotation around the Z-axis

9.11.2.File Example

```
<?xml version="1.0"?>
<Earth_Explorer_File
```

```
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_INT_STRCFG_0101.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.1">
 <Earth_Explorer_Header>
    </Earth_Explorer_Header>
    <Data_Block type="xml">
         <Satellite_Name>CryoSat</Satellite_Name>
         <Mispointing>
            <Aberration_Correction>Yes</Aberration_Correction>
            <Star_Trackers_Limits>
               <Max_Penalty>5</Max_Penalty>
               <Quaternion_Norm_Threshold>0.000001</Quaternion_Norm_Threshold>
               <Max_Time_Gap unit="s">600</Max_Time_Gap>
            </Star_Trackers_Limits>
            <Star_Trackers_Priority>
               <File_Type_1>STR1ATT_0_</File_Type_1>
               <File_Type_2>STR2ATT_0_</File_Type_2>
               <File_Type_3>STR3ATT_0_</File_Type_3>
            </Star_Trackers_Priority>
            <!-- Antenna Bench To Star Tracker rotation angles -->
            <List_of_Star_Trackers count="3">
               <Star_Tracker>
                  <Pre_Launch_Angles>
                     <X_Rotation unit="deg">0.000</X_Rotation>
                     <Y_Rotation unit="deg">0.000</Y_Rotation>
                     <Z_Rotation unit="deg">0.000</Z_Rotation>
                  </Pre_Launch_Angles>
                  <Post_Launch_Misalignment>
                     <X_Rotation unit="deg">0.000</X_Rotation>
                     <Y_Rotation unit="deg">0.000</Y_Rotation>
                     <Z_Rotation unit="deg">0.000</Z_Rotation>
                  </Post_Launch_Misalignment>
               </Star_Tracker>
               <Star_Tracker>
                  <Pre_Launch_Angles>
                     <X_Rotation unit="deg">65.000</X_Rotation>
                     <Y_Rotation unit="deg">0.000</Y_Rotation>
                     <Z_Rotation unit="deg">0.000</Z_Rotation>
                  </Pre_Launch_Angles>
                  <Post_Launch_Misalignment>
                     <X_Rotation unit="deg">0.000</X_Rotation>
                     <Y_Rotation unit="deg">0.000</Y_Rotation>
                     <Z_Rotation unit="deg">0.000</Z_Rotation>
                  </Post_Launch_Misalignment>
               </Star_Tracker>
               <Star_Tracker>
                  <Pre_Launch_Angles>
                     <X_Rotation unit="deg">295.000</X_Rotation>
                     <Y_Rotation unit="deg">0.000</Y_Rotation>
                     <Z_Rotation unit="deg">0.000</Z_Rotation>
                  </Pre_Launch_Angles>
                  <Post_Launch_Misalignment>
                     <X_Rotation unit="deg">0.000</X_Rotation>
                     <Y_Rotation unit="deg">0.000</Y_Rotation>
                     <Z_Rotation unit="deg">0.000</Z_Rotation>
                  </Post_Launch_Misalignment>
               </Star_Tracker>
            </List_of_Star_Trackers>
             <!-- End Antenna Bench To Star Tracker rotation angles -->
         <Satellite_Mechanical_To_Antenna_Bench>
         <Pre_Launch_Angles>
            <X_Rotation unit="deg">0.000</X_Rotation>
            <Y_Rotation unit="deg">354.000</Y_Rotation>
            <Z_Rotation unit="deg">0.000</Z_Rotation>
         </Pre_Launch_Angles>
         <Post_Launch_Misalignment>
            <X_Rotation unit="deg">0.000</X_Rotation>
            <Y_Rotation unit="deg">0.000</Y_Rotation>
            <Z_Rotation unit="deg">0.000</Z_Rotation>
         </Post_Launch_Misalignment>
         </Satellite_Mechanical_To_Antenna_Bench>
         <Satellite_Control_To_Satellite_Mechanical>
            <X_Rotation unit="deg">0.000</X_Rotation>
            <Y_Rotation unit="deg">6.000</Y_Rotation>
            <Z_Rotation unit="deg">0.000</Z_Rotation>
         </Satellite_Control_To_Satellite_Mechanical>
         <Satellite_Attitude_To_Satellite_Control>
            <X_Rotation unit="deg">0.000</X_Rotation>
            <Y_Rotation unit="deg">0.000</Y_Rotation>
            <Z_Rotation unit="deg">270.000</Z_Rotation>
         </Satellite_Attitude_To_Satellite_Control>
      </Mispointing>
      [...]
</Data_Block>
         </Earth_Explorer_File>
```

9.12.DEM Configuration File

9.12.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 163: DEM Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
DEM	Structure (see Table 164)	-	-	Structure containing the DEM model.

Table 164: DEM Configuration File. DEM Structure

Tag name	type	Attribute	C Format	Description
DEM_User_Parameters	Structure (see Table 165)	-	-	Structure containing the User parameters
DEM_Metadata	Structure (see Table 166)	-	-	Structure containing the DEM Metadata.

Table 165: DEM Configuration File. DEM_User_Parameters Structure

Tag name	type	Attribute	C Format	Description
Directory	string	-	%s	Directory where the DEM files are placed. It can be an absolute or a relative path. All the files are assumed to be in the same directory. The supported DEM files can be downloaded from: http://eop-cfi.esa.int/eo_cfi_distribution/DEM/
Cache_Type	string	-	%s	Type of cache used for DEM computations. Possible values: • NO_CACHE • PRELOAD_CACHE • FIFO_CACHE

Cache_Max_Size	integer	size="MB"	%d	Maximum size of memory cache
MiniTiles_Configuration	Structure (see Table 167)	-	-	Mini tile configuration for DEM maximum altitude algorithm
Geoid_Computation	string	-	%s	Flag to indicate if geoid correction must be performed or not in DEM computations. Possible values: • Enabled • Disabled
Geoid_Nof_Harmonics	integer	-	%d	Number of harmonics to be used in geoid correction computation.

Table 166: DEM Configuration File. DEM_Metadata Structure

Tag name	type	Attribute	C Format	Description
Dataset_Model	String	-	%s	Supported dataset models (see http://eop- cfi.esa.int/eo_cfi_distribution/DEM/) DEM model: • GETASSE30_V1 • GETASSE30_V2 • ACE2_9SEC
Description	String	-	%s	DEM description

Table 167: DEM Mini tile configuration. DEM_User_Parameters Structure

Tag name	type	Attribute	C Format	Description
Filename	String	-	%s	Filename or path of the maximum altitude binary file.
Lon_Size	Real	unit="deg"	%d	Longitude size of mini tiles

Lat_Size	Real	unit=”deg”	%d	Latitude size of mini tiles

9.12.2.File Example

```
<?xml version = "1.0" encoding = "UTF-8"?>
<Earth_Explorer_File xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://eop-cfi.esa.int/CFI http://eop-
cfi.esa.int/CFI/EE_CFI_SCHEMAS/EO_OPER_INT_DEMCFG_0103.XSD" xmlns="http://eop-cfi.esa.int/CFI"
schemaVersion="1.3">
   <Earth_Explorer_Header>
     <Fixed_Header>
       <File_Name>CS_TEST_INT_DEMCFG_00000000T000000_99999999T999999_0003</File_Name>
       <File_Description>DEM Configuration File</File_Description>
       <Notes></Notes>
       <Mission>CryoSat</Mission>
       <File_Class>TEST</File_Class>
       <File_Type>INT_DEMCFG</File_Type>
       <Validity_Period>
         <Validity_Start>UTC=0000-00-00T00:00:00</Validity_Start>
         <Validity_Stop>UTC=9999-99-99T99:99:99</Validity_Stop>
       </Validity_Period>
       <File_Version>0001</File_Version>
       <Source>
         <System>What system?</System>
         <Creator>Earth Explorer CFI</Creator>
         <Creator_Version>1.1</Creator_Version>
         <Creation_Date>UTC=2006-07-01T18:11:45</Creation_Date>
       </Source>
     </Fixed_Header>
     <Variable_Header></Variable_Header>
   </Earth_Explorer_Header>
   <Data_Block type="xml">
     <DEM>
       <DEM_User_Parameters>
         <Directory>../../data/ACE2_9SEC</Directory>
         <Cache_Type>FIFO_CACHE</Cache_Type>
         <Cache_Max_Size size=”MB”>2048</Cache_Max_Size>
       </DEM_User_Parameters>
       <DEM_Metadata>
         <Dataset_Model>ACE2_9SEC</Dataset_Model>
         <Description></Description>
       </DEM_Metadata>
     </DEM>
   </Data_Block>
 </Earth_Explorer_File>
```

9.13.Swath Definition File

9.13.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty.
3. Data Block: It consists in a set of structures described in the tables below.

Table 168: Swath Definition File. Data Block

Tag name	type	Attribute	C Format	Description
Swath	Structure (see Table 169)	-	-	Swath structure

Table 169: Swath Definition File. Swath

Tag name		type	Attribute	C Format	Description
Output_File_Description		string	-	%s	File Despcrition for the output swath template file
Output_File_Type		string	-	%s	File type for the output swath template file. It should have the fixed value “MPL_SWTREF”
Swath_Type		string	-	%s	Swath type. It can have one of the following values: • point • line • inertial
Num_Swath_Records		integer	-	%d	Number of points in the swath template file (>0)
Refraction		Structure (See Table 170)	-	-	Refraction model structure
Geometry Union:Choice between the following structures:	List_of_Swath_Points	List of swath geometry (see Table 171)	count=n (number f points in the instantaneous swath, n>=1)	-	List with the geometry data for the calculation of every point in the instantaneous swath.
	Asar_Geometry	Structure (defined in Table 175)	-	-	ASAR geomery
Sat_Nominal_Att		Structure (see Table 176)	-	-	Satellite Nominal Attitude initialization data
Sat_Att		Structure (see Table 177)	-	-	Satellite Attitude initialization data
Instr_Att		Structure (see Table 177)	-	-	Instrument Attitude initialization data

Table 170: Swath Definition File. Refraction

Tag name	type	Attribute	C Format	Description

Model	string	-	%s	Atmospheric refraction model. It can be one of: • NO_REF • STD_REF • USER_REF • PRED_REF
Freq	real	unit="MHz"	%f	Signal Frequency (≥ 0)

Table 171: Swath Definition File. Swath Point

Tag name		type	Attribute	C Format	Description
Swath Point		Union	-	-	Swath point union. It includes one ot hte tags described in the row below.
Swath_Point Union: Choice between the following structures	Pointing_Geometry	Structure (defined in table 183)	-	-	Swath geometry according to
	Distance_Geometry	Structure (defined in Table 174)	-	-	
	Limb_Geometry	Structure (defined in Table 173)	-	-	
	Inertial_Geometry	Structure (defined in Table 173)	-	-	
	Sub Satellite Geometry	empty tag	-	-	

Table 172 Swath Definition File. Point Geometry

Tag name	type	Attribute	C Format	Description
Azimuth	real	unit="deg"	%f	Swath azimuth (-360, 360)
Elevation	real	unit="deg"	%f	Swath elevation (-90, 90)
Altitude	real	unit="m"	%f	Swath altitude (> 0)

Table 173 Swath Definition File. Limb and Inertial Geometry

Tag name	type	Attribute	C Format	Description
Azimuth	real	unit="deg"	%f	Swath azimuth (-360, 360)
Altitude	real	unit="m"	%f	Swath altitude

Table 174: Swath Definition File. Distance Point Geometry

Tag name	type	Attribute	C Format	Description
Azimuth	real	unit=”deg”	%f	Swath azimuth (-360, 360)
Elevation	real	unit=”deg”	%f	Swath elevation (-90, 90)
Altitude	real	unit=”m”	%f	Swath altitude
Distance	real	unit=”m”	%f	Distance

Table 175: Swath Definition File. ASAR Geometry

Tag name		type	Attribute	C Format	Description
Left_Pt		Structure (defined in Table 172)	-	-	Left Point
Mid_Pt		Structure (defined in Table 172)	-	-	Middle Point
Right_Pt		Structure (defined in Table 172)	-	-	Right Point
Choice between one of the following tags:	Narrow_Asar	Structure (defined in Table 191)	-	-	Narrow Asar
	Wide_Asar	Structure (defined in Table 192)	-	-	Wide Asar

Table 176: Swath Definition File. Satellite Nominal Attitude

Tag name		type	Attribute	C Format	Description
	None	Null (no value needed for this tag)	-	-	The satellite nominal attitude frame is not defined.
	AOCS_Model	integer	-	%d	AOCS model
	Parameter_Model	Structure (see Table 178)	-	-	Attitude initialization with parameter model
	Harmonic_Model	Structure (see Table 179)	-	-	Attitude initialization with harmonic model
	File_Model	Structure (see Table 180)	-	-	Attitude initialization with a data file

Table 177: Swath Definition File. Satellite and Instrument Attitude

Tag name	type	Attribute	C Format	Description

Choice between one of the following tags:	None	Null (no value needed for this tag)	-	-	The attitude frame is not defined.
	Harmonic_Model	Structure (see Table 179)	-	-	Attitude initialization with harmonic model
	File_Model	Structure (see Table 180)	-	-	Attitude initialization with a data file
	Angle_Model	Structure (see Table 181)	-	-	Attitude initialization with angles
	Matrix_Model	Structure (see Table 182)	-	-	Attitude initialization with a Matrix

Table 178: Swath Definition File. Parameter Model

Tag name	type	Attribute	C Format	Description
Model	integer	-	%d	Parameter model
List_of_Parameters	List of <Parameter> structures (see Table 183)	count="n" where n is the number of elements in the list	-	List of parameters as used for the CFI function xp_sat_nominal_att_init_model (See [POINT_SUM])

Table 179: Swath Definition File. Harmonic Model

Tag name	type	Attribute	C Format	Description
Angle_Type	integer	-	%d	Angle type
List_of_Harmonics_Pitch	List of <Harmonic_Pitch> structures (see Table 184)	count="n" where n is the number of elements in the list	-	List of harmonic pitch coeficients
List_of_Harmonic_Roll	List of <Harmonic_Roll> structures (see Table 184)	count="n" where n is the number of ele- ments in the list	-	List of harmonic roll coeficients
List_of_Harmonic_Yaw	List of <Harmonic_Yaw> structures (seeTable 184)	count="n" where n is the number of elements in the list	-	List of harmonic yaw coeficients
Offsets (only for Instr_Att element)	Structure (see Table 186)	-	-	Offsets

Table 180: Swath Definition File. File Model

Tag name	type	Attribute	C Format	Description

List_of_Files	List of <File> (see Table 187)	count="n" where n is the number of elements in the list	-	Attitude file list
Auxiliary_File (only for Sat_Att)	string	-	%s	Attitude auxiliary filename (complete path)
Time_Selection	Structure (see Table 188)	-	-	It indicates the time window to be read from the attitude files.

Table 181: Swath Definition File. Angle Model

Tag name	type	Attribute	C Format	Description
Angle_1	real	unit="deg"	%f	Pitch Mispointing angle
Angle_2	real	unit="deg"	%f	Roll Mispointing angle
Angle_3	real	unit="deg"	%f	Yaw Mispointing angle
Offsets (only for Instr_Att element)	Structure (see Table 186)	-	-	Offsets

Table 182: Swath Definition File. Matrix Model

Tag name	type	Attribute	C Format	Description
Row_1	Structure (see Table 190)	-	-	Mispointing matrix first row
Row_2	Structure (see Table 190)	-	-	Mispointing matrix second row
Row_3	Structure (see Table 190)	-	-	Mispointing matrix third row
Offsets (only for Instr_Att element)	Structure (see Table 186)	-	-	Offsets

Table 183: Swath Definition File. List_of_Parameters

Tag name	type	Attribute	C Format	Description
Parameter	string	-	%s	Parameter

Table 184: Swath Definition File. List_of_Harmonics_Pitch/Roll/Yaw

Tag name	type	Attribute	C Format	Description
Harmonic	Structure (see Table 185)	-	-	Harmonic structure

Table 185: Swath Definition File. Harmonic

Tag name	type	Attribute	C Format	Description
Harmonic_Type	integer	-	%d	Harmonic type
Harmonic_Coef	real	-	%f	Harmonic coefficient

Table 186: Swath Definition File. Offsets

Tag name	type	Attribute	C Format	Description
Offset_X	real	unit=”m”	%f	X Offset
Offset_Y	real	unit=”m”	%f	Y Offset
Offset_Z	real	unit=”m”	%f	Z Offset

Table 187: Swath Definition File. File

Tag name	type	Attribute	C Format	Description
File	string	-	-	Attitude filename (complete path)

Table 188: Swath Definition File. Time Selection

Tag name	type	Attribute	C Format	Description
Select_File	Null (no value needed for this tag)	-	-	The whole files will be read from the files
Time_Window	Structure (see Table 189)	-	-	A time window will be read from the files

Table 189: Swath Definition File. Time_Window

Tag name	type	Attribute	C Format	Description
Time_0	real	-	%f	Start time
Time_1	real	-	%f	Stop time

Table 190 Swath Definition File. Row

Tag name	type	Attribute	C Format	Description
Column_1	real	-	%f	Matrix element in the first column

Column_2	real	-	%f	Matrix element in the second column
Column_3	real	-	%f	Matrix element in the third column

Table 191: Swath Definition File. Narrow Asar

Tag name	type	Attribute	C Format	Description
Slant_Range_Left	real	unit=”10e-6s”	%f	Slant Range Extension parameter for left point.

Table 192: Swath Definition File. Wide Asar

Tag name	type	Attribute	C Format	Description
Slant_Range_Left	real	unit=”10e-6s”	%f	Slant Range Extension parameter for left point.
Slant_Range_Right	real	unit=”10e-6s”	%f	Slant Range Extension parameter for right point.

9.13.2.File Example

```
<?xml version="1.0"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_MPL_SW_DEF_0201.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="2.1">

    <Earth_Explorer_Header>
       <Fixed_Header>
         <File_Name>SWATH_DEF_FILE.XML</File_Name>
         <File_Description>Swath Definition File</File_Description>
         <Notes/>
         <Mission>XXXXX</Mission>
         <File_Class>TEST</File_Class>
         <File_Type>MPL_SW_DEF</File_Type>
         <Validity_Period>
            <Validity_Start>UTC=0000-00-00T00:00:00.000000</Validity_Start>
            <Validity_Stop>UTC=9999-99-99T99:99:99.999999</Validity_Stop>
         </Validity_Period>
         <File_Version>1</File_Version>
         <Source>
            <System>CFI Acceptance</System>
            <Creator></Creator>
            <Creator_Version></Creator_Version>
            <Creation_Date>UTC=2003-11-28T17:25:44</Creation_Date>
         </Source>
       </Fixed_Header>
      <Variable_Header/>
```

```
</Earth_Explorer_Header>
<Data_Block type="xml">
  <Swath>
    <Output_File_Description>Area swath</Output_File_Description>
    <Output_File_Type>MPL_SWTREF</Output_File_Type>
    <Swath_Type>closed</Swath_Type>
    <Num_Swath_Records>1200</Num_Swath_Records>
     <Refraction>
       <Model>NO_REF</Model>
      <Freq unit="MHz">000440000000</Freq>
    </Refraction>
    <List_Of_Swath_Points count=”4”>
      <Swath_Point>
          <Pointing_Geometry>
           <Azimuth unit="deg">+270.000000</Azimuth>
           <Elevation unit="deg">+055.750000</Elevation>
           <Altitude unit="m">+000000.000</Altitude>
          </Pointing_Geometry>
      </Swath_Point>


      <Swath_Point>
          <Distance_Geometry>
           <Azimuth unit="deg">+090.000000</Azimuth>
           <Elevation unit="deg">+055.750000</Elevation>
           <Altitude unit="m">+000000.000</Altitude>
           <Distance unit="m">+000001.000</Altitude>
          </Distance_Geometry>
      </Swath_Point>


      <Swath_Point>
          <Pointing_Geometry>
           <Azimuth unit="deg">+180.000000</Azimuth>
           <Elevation unit="deg">+055.750000</Elevation>
           <Altitude unit="m">+000000.000</Altitude>
          </Pointing_Geometry>
      </Swath_Point>



      <Swath_Point>
          <Sub_Satellite_Geometry>
          </Sub_Satellite_Geometry>
      </Swath_Point>


    </List_Of_Swath_Points>


     <Sat_Nominal_Att>

      <Parameter_Model>
        <Model>1</Model>
        <List_of_Parameters count="3">
          <Parameter>-000.167200</Parameter>
          <Parameter>+000.050100</Parameter>
          <Parameter>+003.928400</Parameter>
        </List_of_Parameters>
      </Parameter_Model>
     </Sat_Nominal_Att>
    <Sat_Att>
      <Angle_Model>
        <Angle_1 unit=”deg”>0</Angle_1>
```

```
            <Angle_2 unit=”deg”>0</Angle_2>
            <Angle_3 unit=”deg”>0</Angle_3>
          </Angle_Model>
        </Sat_Att>
        <Instr_Att>
          <None></None>
        </Instr_Att>
       </Swath>
    </Data_Block>
</Earth_Explorer_File>
```

9.14.Swath Template File

9.14.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: It consists in a set of structures described in the tables below.

Table 193: Swath Template File. Variable_Header

Tag name		type	Attribute	C Format	Description
Reference_OSF		string	-	%s	Orbit Scenario File used for generating the file
Reference_SDFSwath_Def_File		string	-	%s	Swath definition file used for generating the file
Absolute_Orbit		integer	-	%ld	Orbit for which the STF has been generated
Start_Validity_Orbit		integer	-	%ld	First orbit for which the STF is valid
Stop_Validity_Orbit		integer	-	%ld	Last orbit for which the STF is valid
Swath_Type		string. It can have one of the following values: • open • closed	-	%s	Swath type
Swath_Point_Type		string. It can have one of the following values: • geodetic • inertial	-	%s	Describes the type of swath points: inertial (RA and Declination) or geodetic (longitude and latitude)
One of the following options:	Orbit_Geometry	Structure (see Table 195)	-	-	Set of orbital parameters
	Orbit_State_Vector	Structure (see Table 196)	-	-	Orbit state vector
Time_Step		real	unit=”s”	%f	Seconds between two swath points
List_of_STF_Altitudes		List of <STF Altitude> (see Table 194)	count=”n” where n is the number of elements in the list (n>=1)	-	
Refraction		Structure (see Table 198)	-	-	Refraction data

Table 194: Swath Template File. STF_Altitude

Tag name	type	Attribute	C Format	Description
STF_Altitude	real	unit=”m”	%f	Altitude for a swath point

Table 195: Swath Template File. Orbit_Geometry

Tag name	type	Attribute	C Format	Description
Repeat_Cycle	real	unit=”day”	%f	Repeat cycle in days
Cycle_Length	real	unit=”orbit”	%f	Cycle length in orbits
MLST_Drift	real	unit=”s/day”	%f	Mean local solar time drift

Table 196: Swath Template File. Orbit_State_Vector

Tag name	type	Attribute	C Format	Description
Absolute_Orbit	integer	-	%d	Orbit number for the swath template file
Pos_X	real	unit=”m”	%f	Position in X coordinate (meters)
Pos_Y	real	unit=”m”	%f	Position in Ycoordinate (meters)
Pos_Z	real	unit=”m”	%f	Position in Z coordinate (meters)
Vel_X	real	unit=”m/s”	%f	Velocity in X coordinate
Vel_Y	real	unit=”m/s”	%f	Velocity in Y coordinate
Vel_Z	real	unit=”m/s”	%f	Velocity in Z coordinate

Table 197: Swath Template File. Line_Altitude

Tag name	type	Attribute	C Format	Description
Left_Altitude	real	unit=”m”	%f	Swath altitude for the left point
Mid_Altitude	real	unit=”m”	%f	Swath altitude for the middle point
Right_Altitude	real	unit=”m”	%f	Swath altitude for the right point

Table 198: Swath Template File. Refraction

Tag name	type	Attribute	C Format	Description
Model	string	-	%s	Atmospheric refraction model. It can be one of: • NO_REF • STD_REF • USER_REF • PRED_REF

Freq	real	unit="MHz"	%f	Signal Frequency (≥ 0)

3. Data Block: It consists in a set of structures described in the tables below.

Table 199: Swath Template File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_STF_Pts	List of <STF_Pt> (See Table 200)	count="n" where n is the number of elements in the list	-	List of points in the swath

Table 200: Swath Template File. STF_Pt

Tag name		type	Attribute	C Format	Description
One of the following options	List_of_Geodetic_Pts	List of <Geodetic_Pt> (see Table 201)	count="n" where n is the number of elements in the list	-	List of records in the swath
	List_of_Inertial_Pts	List_of_Inertial_Pts (see Table 202)			

Table 201: Swath Template File. Geodetic_Pt

Tag name	type	Attribute	C Format	Description
Long	real	unit="deg"	%f	Longitude of the point
Lat	real	unit="deg"	%f	Latitude of the point

Table 202: Swath Template File. Inertial_Pt

Tag name	type	Attribute	C Format	Description
Ra	real	unit="deg"	%f	Right Ascension
Dec	real	unit="deg"	%f	Declination

9.14.2. File Example

```
<?xml version = "1.0" encoding = "UTF-8"?>
<Earth_Explorer_File
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_MPL_SWTREF_0200.XSD"
 xmlns="http://eop-cfi.esa.int/CFI"
```

```
schemaVersion="02.00">
  <Earth_Explorer_Header>
   <Fixed_Header>
        <File_Name>LINE_SWATH_FILE.XML</File_Name>
        <File_Description>Swath Template File</File_Description>
        <Notes/>
        <Mission>XXXXX</Mission>
        <File_Class>TEST</File_Class>
        <File_Type>MPL_SWTDEF</File_Type>
        <Validity_Period>
           <Validity_Start>UTC=0000-00-00T00:00:00.000000</Validity_Start>
           <Validity_Stop>UTC=9999-99-99T99:99:99.999999</Validity_Stop>
        </Validity_Period>
        <File_Version>1</File_Version>
        <Source>
           <System>CFI Acceptance</System>
           <Creator></Creator>
           <Creator_Version></Creator_Version>
           <Creation_Date>UTC=2005-07-09T09:25:44</Creation_Date>
        </Source>
   </Fixed_Header>
    <Variable_Header>
     <Reference_OSF>N/A</Reference_OSF>
     <Reference_SDF>N/A</Reference_SDF>
     <Absolute_Orbit>2000</Absolute_Orbit>
     <Start_Validity_Orbit>1990</Start_Validity_Orbit>
     <Stop_Validity_Orbit>2010</Stop_Validity_Orbit>
     <Swath_Type>open</Swath_Type>
       <Swath_Point_Type>geodetic</Swath_Point_Type>
     <Orbit_Geometry>
       <Repeat_Cycle unit="day">35</Repeat_Cycle>
       <Cycle_Length unit="orbit">501</Cycle_Length>
       <MLST_Drift unit="s/day">+000.000000</MLST_Drift>
     </Orbit_Geometry>
     <Time_Step unit="s">5.029940120</Time_Step>
     <List_of_STF_Altitudes count=”4”>
       <STF_Altitude unit="m">+000000.000</STF_Altitude>
       <STF_Altitude unit="m">+000000.000</STF_Altitude>
       <STF_Altitude unit="m">+000000.000</STF_Altitude>
       <STF_Altitude unit="m">+000000.000</STF_Altitude>
     </List_of_STF_Altitudes>
     <Refraction>
       <Model>NO_REF</Model>
       <Freq unit="MHz">0440000000</Freq>
     </Refraction>
    </Variable_Header>
  </Earth_Explorer_Header>
  <Data_Block type="xml">
   <List_of_STF_Pts count="1200">
     <STF_Pt>
       <List_of_Geodetic_Pts count=”4”>
        <Geodetic_Pt>
         <Long unit="deg">-000.000000</Long>
         <Lat unit="deg">-000.000000</Lat>
        </Geodetic_Pt>
        <Geodetic_Pt>
         <Long unit="deg">-000.000000</Long>
         <Lat unit="deg">-010.000000</Lat>
        </Geodetic_Pt>
        <Geodetic_Pt>
         <Long unit="deg">-010.000000</Long>
```

```
          <Lat unit="deg">-010.000000</Lat>
         </Geodetic_Pt>
         <Geodetic_Pt>
          <Long unit="deg">-010.000000</Long>
          <Lat unit="deg">-000.000000</Lat>
         </Geodetic_Pt>
        </List_of_Geodetic_Pts count=”4”>
        </STF_Pt>

      [...]

        <STF_Pt>
        <List_of_Geodetic_Pts count=”4”>
         <Geodetic_Pt>
          <Long unit="deg">010.000000</Long>
          <Lat unit="deg">350.000000</Lat>
         </Geodetic_Pt>
         <Geodetic_Pt>
          <Long unit="deg">010.000000</Long>
          <Lat unit="deg">000.000000</Lat>
         </Geodetic_Pt>
         <Geodetic_Pt>
          <Long unit="deg">020.000000</Long>
          <Lat unit="deg">-010.000000</Lat>
         </Geodetic_Pt>
         <Geodetic_Pt>
          <Long unit="deg">020.000000</Long>
          <Lat unit="deg">350.000000</Lat>
         </Geodetic_Pt>
        </List_of_Geodetic_Pts>
        </STF_Pt>
        </List_of_STF_Pts>
  </Data_Block>
</Earth_Explorer_File>
```

9.15.Zone Database File

9.15.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 203: Zone Database File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_Zones	List of <Zone> Structures (See Table 204)	count=”n” where n is the number of elements in the list	-	List of zones

Table 204: Zone Database File. Zone

Tag name	type	Attribute	C Format	Description
Zone_Id	string	-	%s	Zone name
Zone_Description	string	-	%s	Zone description
Surface	string	-	%s	Type of surface
Projection	string	-	%s	Projection
Creator	string	-	%s	Cretator name
List_of_Polygon_Pts	list of structures <Polygon_Pt> (See Table 205)	count=”n” where n is the number of ele- ments in the list	-	List of points defining the zone.
Diameter	real	unit=”m”	%f	Diameter of the zone if the list of polygon points is emtpy.

Table 205: Zone Database File. Polygon_Pt

Tag name	type	Attribute	C Format	Description
Long	real	unit=”deg”	%f	longitude of the point (-360, 360)
Lat	real	unit=”deg”	%f	latitude of the point (-90, 90)

9.15.2.File Example

```
<?xml version="1.0"?>
<Earth_Explorer_File
 xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_MPL_ZON_DB_0101.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.1">

   <Earth_Explorer_Header>
      <Fixed_Header>
         <File_Name>ZONE_FILE.XML</File_Name>
         <File_Description>Zone Database File</File_Description>
         <Notes/>
         <Mission>XXXXX</Mission>
         <File_Class>TEST</File_Class>
         <File_Type></File_Type>
         <Validity_Period>
            <Validity_Start>UTC=0000-00-00T00:00:00.000000</Validity_Start>
            <Validity_Stop>UTC=9999-99-99T99:99:99.999999</Validity_Stop>
         </Validity_Period>
         <File_Version>1</File_Version>
         <Source>
            <System>CFI Acceptance</System>
            <Creator></Creator>
            <Creator_Version></Creator_Version>
            <Creation_Date>UTC=2003-11-28T17:25:44</Creation_Date>
         </Source>
```

```
  </Fixed_Header>
  <Variable_Header/>
</Earth_Explorer_Header>
<Data_Block type="xml">
  <List_of_Zones count="5">
  <Zone>
    <Zone_Id>ZMIK____</Zone_Id>
    <Zone_Description></Zone_Description>
    <Surface></Surface>
    <Projection>ANY</Projection>
    <Creator>TEST DATA</Creator>
    <List_of_Polygon_Pts count="003">
      <Polygon_Pt>
        <Long unit="deg">+000.000000</Long>
        <Lat unit="deg">+000.000000</Lat>
      </Polygon_Pt>
      <Polygon_Pt>
        <Long unit="deg">+000.000000</Long>
        <Lat unit="deg">+000.000000</Lat>
      </Polygon_Pt>
      <Polygon_Pt>
        <Long unit="deg">+000.000000</Long>
        <Lat unit="deg">+000.000000</Lat>
      </Polygon_Pt>
    </List_of_Polygon_Pts>
    <Diameter unit="m">+0000000.000</Diameter>
  </Zone>


  <Zone>
    <Zone_Id>SEGMENT_</Zone_Id>
    <Zone_Description></Zone_Description>
    <Surface></Surface>
    <Projection>ANY</Projection>
    <Creator>TEST DATA</Creator>
    <List_of_Polygon_Pts count="002">
      <Polygon_Pt>
        <Long unit="deg">+000.000000</Long>
        <Lat unit="deg">+030.000000</Lat>
      </Polygon_Pt>
      <Polygon_Pt>
        <Long unit="deg">+150.000000</Long>
        <Lat unit="deg">+020.000000</Lat>
      </Polygon_Pt>
    </List_of_Polygon_Pts>
    <Diameter unit="m">+0000000.000</Diameter>
  </Zone>


   <Zone>
    <Zone_Id>POINT_DI</Zone_Id>
    <Zone_Description></Zone_Description>
    <Surface></Surface>
    <Projection>ANY</Projection>
    <Creator>TEST DATA</Creator>
    <List_of_Polygon_Pts count="001">
      <Polygon_Pt>
         <Long unit="deg">+000.000000</Long>
         <Lat unit="deg">+030.000000</Lat>
      </Polygon_Pt>
    </List_of_Polygon_Pts>
    <Diameter unit="m">+0100000.000</Diameter>
```

```
      </Zone>

       <Zone>
        <Zone_Id>POINT___</Zone_Id>
        <Zone_Description></Zone_Description>
        <Surface></Surface>
        <Projection>ANY</Projection>
        <Creator>TEST DATA</Creator>
        <List_of_Polygon_Pts count="001">
          <Polygon_Pt>
             <Long unit="deg">+002.278785</Long>
             <Lat unit="deg">-067.992416</Lat>
          </Polygon_Pt>
        </List_of_Polygon_Pts>
        <Diameter unit="m">+0000000.000</Diameter>
      </Zone>

       <Zone>
        <Zone_Id>Z_WORLD_</Zone_Id>
        <Zone_Description></Zone_Description>
        <Surface></Surface>
        <Projection>ANY</Projection>
        <Creator>TEST DATA</Creator>
        <List_of_Polygon_Pts count="000">
        </List_of_Polygon_Pts>
        <Diameter unit="m">+0000000.000</Diameter>
      </Zone>
      </List_of_Zones>
   </Data_Block>
</Earth_Explorer_File>
```

9.16.Station Database File

9.16.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 206: Station Database File. Data Block

Tag name	type	Attribute	C Format	Description
Station_Id	string	-	%s	Station name
Descriptor	string	-	%s	Station description
Antenna	string	-	%s	Antenna band
Frequency (optional)	real	unit="Hz"	%f	Frequency
Purpose	string	-	%s	Purpose
Type	string	-	%s	
Location	Structure (see Table 207)	-	-	Station location
List_of_Spacecrafts	List of <Spacecraft> structures (see Table 208)	count="n" where n is the number of elements in the list		Spacecraft dependant mask type parameters
Default_El	real	unit="deg"	%f	Default elevation
List_of_Mask_Points	list of <Mask_Point> structures (see Table 209)	count="n" where n is the number of elements in the list	-	Mask points

Table 207: Station Database File. Location

Tag name	type	Attribute	C Format	Description
Long	real	unit="deg"	%f	longitude
Lat	real	unit="deg"	%f	Latitude
Alt	real	unit="deg"	%f	Altitude

Table 208: Station Database File. Spacecraft

Tag name	type	Attribute	C Format	Description
Name	string	-	%s	Spacecraft name
Aos_El	real	unit=”deg”	%f	Acquisition of signal elevation
Los_El	real	unit=”deg”	%f	Loss of signal elevation
Mask	string	-	%s	Mask type

Table 209: Station Database File. Mask_Point

Tag name	type	Attribute	C Format	Description
Az	real	unit=”deg”	%f	Azimuth
El	real	unit=”deg”	%f	Elevation

9.16.2.File Example

```
<?xml version="1.0"?>
<Earth_Explorer_File
xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xsi:schemaLocation=”http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_MPL_GND_DB_0101.XSD”
 xmlns="http://eop-cfi.esa.int/CFI"
 schemaVersion="1.1">
    <Earth_Explorer_Header>
       <Fixed_Header>
         <File_Name>STATION_FILE.XML</File_Name>
         <File_Description>Station Database File</File_Description>
         <Notes/>
         <Mission>XXXXX</Mission>
         <File_Class>TEST</File_Class>
         <File_Type></File_Type>
         <Validity_Period>
            <Validity_Start>UTC=0000-00-00T00:00:00.000000</Validity_Start>
            <Validity_Stop>UTC=9999-99-99T99:99:99.999999</Validity_Stop>
         </Validity_Period>
         <File_Version>1</File_Version>
         <Source>
            <System>CFI Acceptance</System>
            <Creator></Creator>
            <Creator_Version></Creator_Version>
            <Creation_Date>UTC=2003-11-28T17:25:44</Creation_Date>
         </Source>
       </Fixed_Header>
      <Variable_Header/>
    </Earth_Explorer_Header>
    <Data_Block type="xml">
     <List_of_Ground_Stations count="n">
       <Ground_Station>
 <Station_id>GKIRUNBX</Station_id>
  <Descriptor>Kiruna (SWEDEN)</Descriptor>
 <Antenna>X-BAND</Antenna>
 <Purpose>GLOBAL + REGIONAL</Purpose>
 <Type></Type>
 <Location>
   <Long unit="deg">+020.964100</Long>
   <Lat unit="deg">+067.857000</Lat>
   <Alt unit="m">+0362.000</Alt>
 </Location>
 <List_of_Spacecrafts count="1">
   <Spacecraft>
     <Name>SMOS</Name>
     <Aos_El unit="deg">+10.0</Aos_El>
     <Los_El unit="deg">+10.0</Los_El>
     <Mask>AOS_LOS_WITH_MASK</Mask>
   </Spacecraft>
 </List_of_Spacecrafts>
 <Default_El unit="deg">+000.000000</Default_El>
 <List_of_Mask_Points count="073">
   <Mask_Point>
     <Az unit="deg">+000.000000</Az>
     <El unit="deg">+001.250000</El>
   </Mask_Point>
   <Mask_Point>
     <Az unit="deg">+004.000000</Az>
     <El unit="deg">+001.150000</El>
   </Mask_Point>
   <Mask_Point>
     <Az unit="deg">+010.000000</Az>
     <El unit="deg">+001.270000</El>
   </Mask_Point>
   [...]
   <Mask_Point>
     <Az unit="deg">+360.000000</Az>
     <El unit="deg">+001.250000</El>
   </Mask_Point>
 </List_of_Mask_Points>
</Ground_Station>
<Ground_Station>
 <Station_id>GAREA__D</Station_id>
 <Descriptor>AREQUIPA (PEROU)</Descriptor>
 <Antenna>DORIS </Antenna>
 <Purpose></Purpose>
 <Type></Type>
 <Location>
   <Long unit="deg">-071.500000</Long>
   <Lat unit="deg">-016.470000</Lat>
   <Alt unit="m">+2494.000</Alt>
 </Location>
 <List_of_Spacecrafts count="2">
   <Spacecraft>
     <Name>SMOS</Name>
     <Aos_El unit="deg">+10.0</Aos_El>
     <Los_El unit="deg">+10.0</Los_El>
     <Mask>AOS_LOS_WITH_MASK</Mask>
   </Spacecraft>
   <Spacecraft>
     <Name>CryoSat</Name>
     <Aos_El unit="deg">+5.0</Aos_El>
     <Los_El unit="deg">+0.0</Los_El>
```

```
            <Mask>AOS_LOS</Mask>
          </Spacecraft>
        </List_of_Spacecrafts>
        <Default_El unit="deg">+012.000000</Default_El>
        <List_of_Mask_Points count="000">
        </List_of_Mask_Points>
      </Ground_Station>
      [...]
   </List_of_Ground_Stations>
  </Data_Block>
</Earth_Explorer_File>
```

9.17.TLE File

The format of the TLE files are described in: http://celestrak.com

9.18.Precise Propagator Configuration File

9.18.1.Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT].
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 210: Precise Propagator Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
Models_Path	string	-	%s	Path where files neccesary for models are looked for.
Gravity_Flag	long integer	-	%ld	Gravity perturbation used (1) or not (0).
Thirdbody_Flag	long integer	-	%ld	Third bodies (Sun and Moon) perturbation used (1) or not (0).
Atmosphere_Flag	long integer	-	%ld	Atmosphere perturbation used (1) or not (0).
Srp_Flag	long integer	-	%ld	Solar radiation pressure perturbation used (1) or not (0).
Time_Step	real	unit=”s”	%lf	Simulation step.
Gravity_File	string	-	%s	File with data of gravitational model.
Gravity_Degree	long integer	-	%ld	Degree used gravity model.
Gravity_Order	long integer	-	%ld	Order used in gravity model.
Sga_Flag	long integer	-	%ld	Parameters used (0) or data read from file (1).
Sga_Ap_File	string	-	%s	File with Geomagnetic Activity index values.
Sga_F107_File	string	-	%s	File with F10.7 Solar Activity index values
AP	real	-	%lf	Geomagnetic Activity Index (daily value).
F107	real	-	%lf	F10.7 Index Solar Activy Index (daily value).
F107A	real	-	%lf	F10.7 Index Solar Activy Index (value averaged over 3 months).
SC_Mass	real	unit=”kg”	%lf	S/C mass.
SC_Drag_Area	real	unit=”m2”	%lf	S/C effective drag area.
SC_Drag_Coef	real	-	%lf	S/C drag coefficient.

SC_Srp_Area	real	unit=”m2”	%lf	S/C effective SRP area.
SC_Srp_Coef	real	-	%lf	S/C SRP coefficient.

9.18.2.File Example

```
 <?xml version="1.0"?>
<Earth_Explorer_File xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLoca-
tion="http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/
EO_OPER_INT_PPRCFG_0100.XSD" schemaVersion="1.0" xmlns="http://eop-cfi.esa.int/CFI">
    <Earth_Explorer_Header>
     <Fixed_Header>
      <File_Name>CS_TEST_INT_PPRCFG_00000000T000000_99999999T999999_0001</File_Name>
      <File_Description>Precise propagation configuration file</File_Description>
      <Notes/>
      <Mission>Cryosat</Mission>
      <File_Class>TEST</File_Class>
      <File_Type>INT_PPRCFG</File_Type>
      <Validity_Period>
        <Validity_Start>UTC=0000-00-00T00:00:00</Validity_Start>
        <Validity_Stop>UTC=9999-99-99T99:99:99</Validity_Stop>
      </Validity_Period>
      <File_Version>0001</File_Version>
      <Source>
        <System>System</System>
        <Creator>Creator</Creator>
        <Creator_Version>3.7.2</Creator_Version>
        <Creation_Date>UTC=2008-10-28T13:00:00</Creation_Date>
      </Source>
    </Fixed_Header>
    <Variable_Header/>
   </Earth_Explorer_Header>
   <Data_Block type="xml">
    <Models_Path>/models_full_path/models</Models_Path>
    <Gravity_Flag>1</Gravity_Flag>
    <Thirdbody_Flag>1</Thirdbody_Flag>
    <Atmosphere_Flag>1</Atmosphere_Flag>
    <Srp_Flag>1</Srp_Flag>
    <Time_Step unit="s">100.000000</Time_Step>
    <Gravity_File>gravity_file.grv</Gravity_File>
    <Gravity_Degree>9</Gravity_Degree>
    <Gravity_Order>8</Gravity_Order>
    <Sga_Flag>1</Sga_Flag>
    <Sga_Ap_File>ap_file.sga</Sga_Ap_File>
    <Sga_F107_File>f107_file.sga</Sga_F107_File>
    <AP>100.000000</AP>
    <F107>30.000000</F107>
    <F107A>29.000000</F107A>
    <SC_Mass unit="kg">2000.000000</SC_Mass>
    <SC_Drag_Area unit="m2">4.000000</SC_Drag_Area>
    <SC_Drag_Coef>2.000000</SC_Drag_Coef>
    <SC_Srp_Area unit="m2">3.000000</SC_Srp_Area>
    <SC_Srp_Coef>1.000000</SC_Srp_Coef>
  </Data_Block>
</Earth_Explorer_File>
```

10.RUNTIME PERFORMANCES

The library performance has been measured by dedicated test procedures run in 5 different platforms under the below specified machines:

OS ID	*Processor*	*OS*	*RAM*
LINUX64	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz	GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)	4 GB
LINUX32_LEGACY	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz	GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)	4 GB
LINUX64_LEGACY	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz	GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)	4 GB
MACIN64	Intel(R) Core(TM)2 Duo @ 2.00GHz	MAC OS X V10.5.8	1 GB
WINDOWS	Intel(R) Core(TM)2 Duo CPU T8100 @ 2.10 GHz	Microsoft Windows XP Professional Version 2002 Service Pack 3	2.99 GB

The table below shows the time (in miliseconds - ms) each function takes to be run under each platform:

Function id	*LINUX64*	*LINUX32 LEGACY*	*LINUX64 LEGACY*	*MACIN64*	*WINDOWS*
xd_read_bulletin	1,400000	2,000000	1,700000	2,400000	32,340000
xd_read_orbit_file * 3 OSVs	0,200000	0,400000	0,200000	0,500000	8,750000
xd_read_fhr	0,100000	0,200000	0,300000	0,300000	7,810000
xd_write_orbit_file * 3 OSVs written	0,400000	0,200000	0,400000	0,900000	3,750000
xd_read_doris_header	0,000000	0,100000	0,100000	0,000000	38,580002
xd_read_doris * 1171 elements	1,100000	1,700000	1,200000	1,800000	45,939999

xd_write_doris * 1171 records written	3,000000	4,000000	3,000000	7,000000	43,799999
xd_read_osf * 5 orbit changes	0,600000	0,600000	0,700000	1,100000	7,340000
xd_write_osf * 5 orbit changes	0,600000	0,500000	0,400000	1,300000	4,070000
xd_read_star_tracker_conf_file * 2000 records read	15,400000	14,300000	14,800000	24,400000	85,610001
xd_read_star_tracker	4,000000	6,200000	3,600000	5,400000	123,419998
xd_read_att * 5 Quaternions	0,300000	0,300000	0,200000	0,400000	5,630000
xd_write_att * 5 Quaternions	0,230000	0,400000	0,400000	0,800000	2,453000
xd_read_precise_propag_file	0,160000	0,250000	0,080000	0,290000	5,484000
xd_free_dem_config_file	0,140000	0,090000	0,110000	0,270000	9,920000
xd_read_dem	129,000000	237,000000	162,000000	231,000000	2701,199951
xd_read_sdf	0,500000	0,400000	0,200000	0,800000	18,440001
xd_read_stf_vhr * 1200 records read	118,300003	98,000000	96,800003	158,500000	587,299988
xd_read_stf	172,100006	135,399994	132,699997	218,199997	742,770020
xd_write_stf * 1200 records written	76,000000	74,000000	74,000000	124,000000	346,899994
xd_read_zone	6,600000	6,200000	7,000000	11,500000	58,900002
xd_read_zone_file * 41 zones, 888 Polygon_Pts	7,400000	7,600000	7,500000	13,000000	47,500000
xd_read_zone_ids * 41 records read	6,700000	6,200000	7,000000	11,500000	42,810001
xd_read_station	12,300000	10,600000	12,200000	18,799999	68,120003

xd_read_station_file * 124 records read	13,000000	13,000000	16,000000	23,000000	73,500000
xd_read_station_id * 124 records read	13,100000	10,700000	11,900000	19,000000	68,430000
xd_read_star	1,800000	1,100000	1,100000	1,600000	21,879999
xd_read_star_file * 1006 stars	197,000000	91,000000	86,000000	140,000000	948,299988
xd_read_star_id * 1006 stars	101,000000	56,000000	61,000000	98,000000	940,599976
xd_xml_validate	0,600000	0,600000	0,600000	1,200000	70,000000

Note that when the value “0.000000” is defined for a function in a certain platform, it means that its running time is lower than 1 nano-second and so it can be considered as “0”.

11.LIBRARY PRECAUTIONS

The following precaution shall be taking into account when using EO_DATA_HANDLING library:

- None

12.KNOWN PROBLEMS

The following precautions shall be taken into account when using the CFI software libraries:

Table 211: Known problems

CFI library	Problem	Work around solution
-	-	-