

Earth Explorer Mission CFI Software

EXPLORER_DATA_HANDLING SOFTWARE USER MANUAL

Code: EE-MA-DMS-GS-0007

Issue: 3.7.3

Date: 07/05/2010

	Name	Function	Signature
Prepared by:	Juan José Borrego Bote	Project Engineer	
Checked by:	José Antonio González Abeytua	Project Manager	
Approved by:	José Antonio González Abeytua	Project Manager	

DEIMOS Space S.L.
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, 2010

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. or ESA.

Document Information

Contract Data		Classification	
Contract Number:	15583/01/NL/GS	Internal	<input type="checkbox"/>
		Public	<input type="checkbox"/>
Contract Issuer:	ESA / ESTEC	Industry	<input checked="" type="checkbox"/>
		Confidential	<input type="checkbox"/>

External Distribution		
Name	Organisation	Copies

Electronic handling	
Word Processor:	Adobe Framemaker 7.0
Archive Code:	P/SUM/DMS/01/026-044
Electronic file name:	ee-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	<ul style="list-style-type: none"> • Maintenance release. • New features: <ul style="list-style-type: none"> - function xd_xml_validate 	26/05/06	
3.6	<ul style="list-style-type: none"> • Maintenance release • New features: <ul style="list-style-type: none"> - Validator function and executable for XML files (xd_xml_validate and xml_validator) 	24/11/06	
3.7	<ul style="list-style-type: none"> • Maintenance release • New features: <ul style="list-style-type: none"> - Function expcfi_check_libs - Library version for MAC OS X on Intel (32 and 64-bits) 	13/07/07	
3.7.2	<ul style="list-style-type: none"> • Maintenance release • New features: <ul style="list-style-type: none"> - Reading and writing functions for TLE - New format for orbit files: reference frame added to the variable header. 	31/07/08	
3.7.3	<ul style="list-style-type: none"> • Maintenance release • New section: List of schema's version • New features: <ul style="list-style-type: none"> - DEM GETASSEv2 supported - IERS B Bulletin format 2010 - Ground Station DB 1.4 supported 	07/05/10	

Table of Contents

1. SCOPE.....	19
2. ACRONYMS AND NOMENCLATURE	20
2.1. Acronyms	20
2.2. Nomenclature	20
3. APPLICABLE AND REFERENCE DOCUMENTS	21
3.1. Applicable Documents	21
3.2. Reference Documents	21
4. INTRODUCTION	22
4.1. Functions Overview	22
4.1.1. Reading routines	22
4.1.2. Writing routines	22
4.1.3. Functions to free memory.....	23
4.1.4. Validation of XML files	23
5. LIBRARY INSTALLATION	24
6. LIBRARY USAGE.....	25
6.1. Usage hints	27
6.2. General Enumerations	27
6.3. Data Structures	32
7. CFI FUNCTIONS DESCRIPTION.....	47
7.1. xd_read_fhr	48
7.1.1. Overview	48
7.1.2. Calling interface	48
7.1.3. Input parameters	48
7.1.4. Output parameters	48
7.1.5. Warnings and errors	49
7.1.6. Runtime performances.....	49
7.2. xd_read_bulletin.....	50
7.2.1. Overview	50
7.2.2. Calling interface	50
7.2.3. Input parameters	50
7.2.4. Output parameters	50
7.2.5. Warnings and errors	51
7.2.6. Runtime performances.....	51
7.3. xd_read_orbit_file	52
7.3.1. Overview	52
7.3.2. Calling interface	52

7.3.3. Input parameters	52
7.3.4. Output parameters	53
7.3.5. Warnings and errors	54
7.3.6. Runtime performances.....	54
7.4. xd_free_orbit_file.....	56
7.4.1. Overview	56
7.4.2. Calling interface	56
7.4.3. Input parameters	56
7.4.4. Output parameters	56
7.5. xd_read_doris	57
7.5.1. Overview	57
7.5.2. Calling interface	57
7.5.3. Input parameters	57
7.5.4. Output parameters	58
7.5.5. Warnings and errors	58
7.5.6. Runtime performances.....	59
7.6. xd_free_doris.....	60
7.6.1. Overview	60
7.6.2. Calling interface	60
7.6.3. Input parameters	60
7.6.4. Output parameters	60
7.7. xd_read_doris_header	61
7.7.1. Overview	61
7.7.2. Calling interface	61
7.7.3. Input parameters	61
7.7.4. Output parameters	61
7.7.5. Warnings and errors	62
7.7.6. Runtime performances.....	63
7.8. xd_read_osf	64
7.8.1. Overview	64
7.8.2. Calling interface	64
7.8.3. Input parameters	64
7.8.4. Output parameters	64
7.8.5. Warnings and errors	65
7.8.6. Runtime performances.....	66
7.9. xd_free_osf.....	67
7.9.1. Overview	67
7.9.2. Calling interface	67
7.9.3. Input parameters	67
7.9.4. Output parameters	67
7.10. xd_read_sdf	68
7.10.1. Overview	68
7.10.2. Calling interface	68
7.10.3. Input parameters	68
7.10.4. Output parameters	69

7.10.5. Warnings and errors	69
7.10.6. Runtime performances.....	70
7.11. xd_free_sdf.....	71
7.11.1. Overview	71
7.11.2. Calling interface	71
7.11.3. Input parameters	71
7.11.4. Output parameters	71
7.12. xd_read_stf.....	72
7.12.1. Overview	72
7.12.2. Calling interface	72
7.12.3. Input parameters	72
7.12.4. Output parameters	72
7.12.5. Warnings and errors	73
7.12.6. Runtime performances.....	74
7.13. xd_free_stf.....	75
7.13.1. Overview	75
7.13.2. Calling interface	75
7.13.3. Input parameters	75
7.13.4. Output parameters	75
7.14. xd_read_stf_vhr.....	76
7.14.1. Overview	76
7.14.2. Calling interface	76
7.14.3. Input parameters	77
7.14.4. Output parameters	77
7.14.5. Warnings and errors	77
7.14.6. Runtime performances.....	78
7.15. xd_free_stf_vhr	79
7.15.1. Overview	79
7.15.2. Calling interface	79
7.15.3. Input parameters	79
7.15.4. Output parameters	79
7.16. xd_read_att.....	80
7.16.1. Overview	80
7.16.2. Calling interface	80
7.16.3. Input parameters	80
7.16.4. Output parameters	80
7.16.5. Warnings and errors	81
7.16.6. Runtime performances.....	82
7.17. xd_free_att.....	83
7.17.1. Overview	83
7.17.2. Calling interface	83
7.17.3. Input parameters	83
7.17.4. Output parameters	83
7.18. xd_read_star_tracker	84
7.18.1. Overview	84

7.18.2. Calling interface	84
7.18.3. Input parameters	84
7.18.4. Output parameters	85
7.18.5. Warnings and errors	85
7.18.6. Runtime performances.....	86
7.19. xd_free_star_tracker.....	87
7.19.1. Overview	87
7.19.2. Calling interface	87
7.19.3. Input parameters	87
7.19.4. Output parameters	87
7.20. xd_read_star_tracker_conf_file.....	88
7.20.1. Overview	88
7.20.2. Calling interface	88
7.20.3. Input parameters	89
7.20.4. Output parameters	89
7.20.5. Warnings and errors	90
7.20.6. Runtime performances.....	90
7.21. xd_read_dem	91
7.21.1. Overview	91
7.21.2. Calling interface	91
7.21.3. Input parameters	91
7.21.4. Output parameters	92
7.21.5. Warnings and errors	92
7.21.6. Runtime performances.....	93
7.22. xd_free_dem.....	94
7.22.1. Overview	94
7.22.2. Calling interface	94
7.22.3. Input parameters	94
7.22.4. Output parameters	94
7.23. xd_read_dem_config_file.....	95
7.23.1. Overview	95
7.23.2. Calling interface	95
7.23.3. Input parameters	95
7.23.4. Output parameters	95
7.23.5. Warnings and errors	96
7.24. xd_read_zone	98
7.24.1. Overview	98
7.24.2. Calling interface	98
7.24.3. Input parameters	98
7.24.4. Output parameters	98
7.24.5. Warnings and errors	99
7.24.6. Runtime performances.....	100
7.25. xd_free_zone	101
7.25.1. Overview	101
7.25.2. Calling interface	101

7.25.3. Input parameters	101
7.25.4. Output parameters	101
7.26. xd_read_zone_file	102
7.26.1. Overview	102
7.26.2. Calling interface	102
7.26.3. Input parameters	102
7.26.4. Output parameters	102
7.26.5. Warnings and errors	103
7.26.6. Runtime performances.....	104
7.27. xd_free_zone_file	105
7.27.1. Overview	105
7.27.2. Calling interface	105
7.27.3. Input parameters	105
7.27.4. Output parameters	105
7.28. xd_read_zone_id	106
7.28.1. Overview	106
7.28.2. Calling interface	106
7.28.3. Input parameters	106
7.28.4. Output parameters	106
7.28.5. Warnings and errors	107
7.28.6. Runtime performances.....	108
7.29. xd_free_zone_id	109
7.29.1. Overview	109
7.29.2. Calling interface	109
7.29.3. Input parameters	109
7.29.4. Output parameters	109
7.30. xd_read_station	110
7.30.1. Overview	110
7.30.2. Calling interface	110
7.30.3. Input parameters	110
7.30.4. Output parameters	110
7.30.5. Warnings and errors	111
7.30.6. Runtime performances.....	112
7.31. xd_read_station_file	113
7.31.1. Overview	113
7.31.2. Calling interface	113
7.31.3. Input parameters	113
7.31.4. Output parameters	113
7.31.5. Warnings and errors	114
7.31.6. Runtime performances.....	115
7.32. xd_free_station_file.....	116
7.32.1. Overview	116
7.32.2. Calling interface	116
7.32.3. Input parameters	116
7.32.4. Output parameters	116

7.33. xd_read_station_id	117
7.33.1. Overview	117
7.33.2. Calling interface	117
7.33.3. Input parameters	117
7.33.4. Output parameters	117
7.33.5. Warnings and errors	118
7.33.6. Runtime performances.....	118
7.34. xd_free_station_id	120
7.34.1. Overview	120
7.34.2. Calling interface	120
7.34.3. Input parameters	120
7.34.4. Output parameters	120
7.35. xd_read_star	121
7.35.1. Overview	121
7.35.2. Calling interface	121
7.35.3. Input parameters	121
7.35.4. Output parameters	121
7.35.5. Warnings and errors	122
7.35.6. Runtime performances.....	122
7.36. xd_read_star_file	123
7.36.1. Overview	123
7.36.2. Calling interface	123
7.36.3. Input parameters	123
7.36.4. Output parameters	123
7.36.5. Warnings and errors	124
7.36.6. Runtime performances.....	124
7.37. xd_read_star_id	125
7.37.1. Overview	125
7.37.2. Calling interface	125
7.37.3. Input parameters	125
7.37.4. Output parameters	125
7.37.5. Warnings and errors	126
7.37.6. Runtime performances.....	126
7.38. xd_read_tle	127
7.38.1. Overview	127
7.38.2. Calling interface	127
7.38.3. Input parameters	128
7.38.4. Output parameters	128
7.38.5. Warnings and errors	129
7.38.6. Runtime performances.....	129
7.39. xd_free_tle.....	130
7.39.1. Overview	130
7.39.2. Calling interface	130
7.39.3. Input parameters	130
7.39.4. Output parameters	130

7.40. xd_write_orbit_file.....	131
7.40.1. Overview	131
7.40.2. Calling interface	131
7.40.3. Input parameters	131
7.40.4. Output parameters	131
7.40.5. Warnings and errors	132
7.40.6. Runtime performances.....	133
7.41. xd_write_osf.....	134
7.41.1. Overview	134
7.41.2. Calling interface	134
7.41.3. Input parameters	134
7.41.4. Output parameters	135
7.41.5. Warnings and errors	135
7.41.6. Runtime performances.....	136
7.42. xd_write_doris.....	137
7.42.1. Overview	137
7.42.2. Calling interface	137
7.42.3. Input parameters	137
7.42.4. Output parameters	137
7.42.5. Warnings and errors	138
7.42.6. Runtime performances.....	138
7.43. xd_write_stf.....	139
7.43.1. Overview	139
7.43.2. Calling interface	139
7.43.3. Input parameters	139
7.43.4. Output parameters	139
7.43.5. Warnings and errors	140
7.43.6. Runtime performances.....	141
7.44. xd_write_att.....	142
7.44.1. Overview	142
7.44.2. Calling interface	142
7.44.3. Input parameters	142
7.44.4. Output parameters	143
7.44.5. Warnings and errors	143
7.44.6. Runtime performances.....	144
7.45. xd_write_tle.....	145
7.45.1. Overview	145
7.45.2. Calling interface	145
7.45.3. Input parameters	145
7.45.4. Output parameters	146
7.45.5. Warnings and errors	146
7.45.6. Runtime performances.....	146
7.46. xd_xml_validate	147
7.46.1. Overview	147
7.46.2. Calling interface	147

7.46.3. Input parameters147
 7.46.4. Output parameters148
 7.46.5. Warnings and errors148
 7.46.6. Runtime performances.....149
 7.46.7. Executable program.....150

8.List of schema's versions151

9. Files Format specification..... 153

9.1. Fixed Header 153
 9.1.1. Format.....153
 9.1.2. Example155
 9.2. Predicted Orbit files 155
 9.2.1. Format.....155
 9.2.2. Example157
 9.3. Restituted Orbit files 158
 9.4. Doris Preliminary/Precise files..... 158
 9.5. Orbit Scenario files..... 158
 9.5.1. Format.....159
 9.5.2. Example160
 9.6. Orbit Event files 161
 9.6.1. Format.....161
 9.6.2. Example162
 9.7. DORIS Navigator files 163
 9.8. Star Tracker files 164
 9.9. Satellite Configuration File 165
 9.9.1. Format.....165
 9.9.2. File Example.....166
 9.10. Attitude File..... 168
 9.10.1. Format.....168
 9.10.2. File Example.....170
 9.11. Star tracker configuration File..... 172
 9.11.1. Format.....172
 9.11.2. File Example.....173
 9.12. DEM Configuration File 176
 9.12.1. Format.....176
 9.12.2. File Example.....177
 9.13. Swath Definition File 179
 9.13.1. Format.....179
 9.13.2. File Example.....185
 9.14. Swath Template File..... 187
 9.14.1. Format.....187
 9.14.2. File Example.....189
 9.15. Zone Database File 192
 9.15.1. Format.....192

9.15.2. File Example.....	192
9.16. Station Database File.....	195
9.16.1. Format.....	195
9.16.2. File Example.....	196
9.17. TLE File	198
10. LIBRARY PRECAUTIONS.....	199
11. KNOWN PROBLEMS.....	200

List of Tables

Table 1:	CFI functions included within EXPLORER_DATA_HANDLING library	26
Table 2:	Enumerations within EXPLORER_DATA_HANDLING library	27
Table 3:	EXPLORER_DATA_HANDLING Structures	33
Table 4:	Input parameters of xd_read_fhr function	48
Table 5:	Output parameters of xd_read_fhr function	48
Table 6:	Error messages of xd_read_fhr function	49
Table 7:	Runtime performances of xd_read_fhr function	49
Table 8:	Input parameters of xd_read_bulletin function	50
Table 9:	Output parameters of xd_read_bulletin function	50
Table 10:	Error messages of xd_read_bulletin function	51
Table 11:	Runtime performances of xd_read_bulletin function	51
Table 12:	Input parameters of xd_read_orbit_file function	52
Table 13:	Output parameters of xd_read_orbit_file function	53
Table 14:	Error messages of xd_read_orbit_file function	54
Table 15:	Runtime performances of xd_read_orbit_file function	55
Table 16:	Input parameters of xd_free_orbit_file function	56
Table 17:	Input parameters of xd_read_doris function	57
Table 18:	Output parameters of xd_read_doris function	58
Table 19:	Error messages of xd_read_doris function	58
Table 20:	Runtime performances of xd_read_doris function	59
Table 21:	Input parameters of xd_free_doris function	60
Table 22:	Input parameters of xd_read_doris_header function	61
Table 23:	Output parameters of xd_read_doris_header function	62
Table 24:	Error messages of xd_read_doris function	62
Table 25:	Runtime performances of xd_read_doris_header function	63
Table 26:	Input parameters of xd_read_osf function	64
Table 27:	Output parameters of xd_read_osf function	64
Table 28:	Error messages of xd_read_osf function	65
Table 29:	Runtime performances of xd_read_osf function	66
Table 30:	Input parameters of xd_free_osf function	67
Table 31:	Input parameters of xd_read_sdf function	68
Table 32:	Output parameters of xd_read_sdf function	69
Table 33:	Error messages of xd_read_sdf function	69
Table 34:	Runtime performances of xd_read_sdf function	70
Table 35:	Input parameters of xd_free_sdf function	71
Table 36:	Input parameters of xd_read_stf function	72
Table 37:	Output parameters of xd_read_stf function	72

Table 38:	Error messages of xd_read_stf function	73
Table 39:	Runtime performances of xd_read_stf function	74
Table 40:	Input parameters of xd_free_stf function.....	75
Table 41:	Input parameters of xd_read_stf_vhr function.....	77
Table 42:	Output parameters of xd_read_stf_vhr function	77
Table 43:	Error messages of xd_read_stf_vhr function	78
Table 44:	Runtime performances of xd_read_stf_vhr function	78
Table 45:	Input parameters of xd_free_stf_vhr function	79
Table 46:	Input parameters of xd_read_att function	80
Table 47:	Output parameters of xd_read_att function	80
Table 48:	Error messages of xd_read_att function	81
Table 49:	Runtime performances of xd_read_att function	82
Table 50:	Input parameters of xd_free_att function.....	83
Table 51:	Input parameters of xd_read_star_tracker function	84
Table 52:	Output parameters of xd_read_star_tracker function	85
Table 53:	Error messages of xd_read_star_tracker function.....	86
Table 54:	Runtime performances of xd_read_star_tracker function.....	86
Table 55:	Input parameters of xd_free_star_tracker function.....	87
Table 56:	Input parameters of xd_read_star_tracker_conf_file function.....	89
Table 57:	Output parameters of xd_read_star_tracker_conf_file function	89
Table 58:	Error messages of xd_read_star_tracker_conf_file function	90
Table 59:	Runtime performances of xd_read_star_tracker_conf_file function	90
Table 60:	Input parameters of xd_read_dem function	91
Table 61:	Output parameters of xd_read_dem function	92
Table 62:	Error messages of xd_read_dem function	92
Table 63:	Runtime performances of xd_read_dem function	93
Table 64:	Input parameters of xd_free_dem function.....	94
Table 65:	Input parameters of xd_read_dem_config_file function	95
Table 66:	Output parameters of xd_read_dem_config_file function.....	96
Table 67:	Error messages of xd_read_dem_config_file function	96
Table 68:	Input parameters of xd_read_zone function	98
Table 69:	Output parameters of xd_read_zone function.....	98
Table 70:	Error messages of xd_read_zone function.....	99
Table 71:	Runtime performances of xd_read_zone function.....	100
Table 72:	Input parameters of xd_free_zone function	101
Table 73:	Input parameters of xd_read_zone_file function	102
Table 74:	Output parameters of xd_read_zone_file function	102
Table 75:	Error messages of xd_read_zone_file function	103
Table 76:	Runtime performances of xd_read_zone_file function	104
Table 77:	Input parameters of xd_free_zone_file function.....	105

Table 78:	Input parameters of xd_read_zone_id function	106
Table 79:	Output parameters of xd_read_zone_id function.....	106
Table 80:	Error messages of xd_read_zone_id function.....	107
Table 81:	Runtime performances of xd_read_zone_id function.....	108
Table 82:	Input parameters of xd_free_zone_id function	109
Table 83:	Input parameters of xd_read_station function	110
Table 84:	Output parameters of xd_read_station function.....	110
Table 85:	Error messages of xd_read_station function.....	111
Table 86:	Runtime performances of xd_read_station function.....	112
Table 87:	Input parameters of xd_read_station_file function	113
Table 88:	Output parameters of xd_read_station_file function	113
Table 89:	Error messages of xd_read_station_file function	114
Table 90:	Runtime performances of xd_read_station_file function	115
Table 91:	Input parameters of xd_free_station_file function.....	116
Table 92:	Input parameters of xd_read_station_id function	117
Table 93:	Output parameters of xd_read_station_id function	117
Table 94:	Error messages of xd_read_station_id function.....	118
Table 95:	Runtime performances of xd_read_station_id function.....	119
Table 96:	Input parameters of xd_free_station_id function.....	120
Table 97:	Input parameters of xd_read_star function	121
Table 98:	Output parameters of xd_read_star function	121
Table 99:	Error messages of xd_read_star function.....	122
Table 100:	Runtime performances of xd_read_star function.....	122
Table 101:	Input parameters of xd_read_star_file function.....	123
Table 102:	Output parameters of xd_read_star_file function	123
Table 103:	Error messages of xd_read_star_file function	124
Table 104:	Runtime performances of xd_read_star_file function	124
Table 105:	Input parameters of xd_read_star_id function	125
Table 106:	Output parameters of xd_read_star_id function	125
Table 107:	Error messages of xd_read_star_id function	126
Table 108:	Runtime performances of xd_read_star_id function	126
Table 109:	Input parameters of xd_read_tle function	128
Table 110:	Output parameters of xd_read_tle function	128
Table 111:	Error messages of xd_read_tle function	129
Table 112:	Runtime performances of xd_read_tle function	129
Table 113:	Input parameters of xd_free_tle function.....	130
Table 114:	Input parameters of xd_write_orbit_file function.....	131
Table 115:	Output parameters of xd_write_orbit_file function	132
Table 116:	Error messages of xd_write_orbit_file function	132
Table 117:	Runtime performances of xd_write_orbit_file function	133

Table 118:	Input parameters of xd_write_osf function.....	134
Table 119:	Output parameters of xd_write_osf function.....	135
Table 120:	Error messages of xd_write_osf function.....	135
Table 121:	Runtime performances of xd_write_osf function.....	136
Table 122:	Input parameters of xd_write_doris function.....	137
Table 123:	Output parameters of xd_write_doris function.....	138
Table 124:	Error messages of xd_write_doris function.....	138
Table 125:	Runtime performances of xd_write_doris function.....	138
Table 126:	Input parameters of xd_write_stf function.....	139
Table 127:	Output parameters of xd_write_stf function.....	140
Table 128:	Error messages of xd_write_stf function.....	140
Table 129:	Runtime performances of xd_write_stf function.....	141
Table 130:	Input parameters of xd_write_att function.....	142
Table 131:	Output parameters of xd_write_att function.....	143
Table 132:	Error messages of xd_write_att function.....	143
Table 133:	Runtime performances of xd_write_att function.....	144
Table 134:	Input parameters of xd_write_tle function.....	145
Table 135:	Output parameters of xd_write_tle function.....	146
Table 136:	Error messages of xd_write_tle function.....	146
Table 137:	Runtime performances of xd_write_tle function.....	146
Table 138:	Input parameters of xd_xml_validate function.....	147
Table 139:	Output parameters of xd_xml_validate function.....	148
Table 140:	Error messages of xd_xml_validate function.....	148
Table 141:	Runtime performances of xd_xml_validate function.....	149
Table 142:	List of schema's versions.....	151
Table 143:	Fixed Header Structure.....	153
Table 144:	Fixed Header. Validity Period.....	154
Table 145:	Fixed Header. Source.....	155
Table 146:	Predicted Orbit File. Variable_Header.....	156
Table 147:	Predicted Orbit File. Data_Block.....	156
Table 148:	Predicted Orbit File. OSV.....	156
Table 149:	Orbit Scenario File. Data_Block.....	159
Table 150:	Orbit Scenario File. Orbit_Change.....	159
Table 151:	Orbit Scenario File. Orbit.....	159
Table 152:	Orbit Scenario File. Cycle.....	159
Table 153:	Orbit Scenario File. Time_of_ANX.....	160
Table 154:	Orbit Scenario File. Data_Block.....	161
Table 155:	Satellite Configuration File. Data Block.....	165
Table 156:	Satellite Configuration File. NORAD_Data Structure.....	165
Table 157:	Satellite Configuration File. Lib_Init Structure.....	165

Table 158:	Satellite Configuration File. Orbit_InitStructure	165
Table 159:	Satellite Configuration File. Low and Tight Tolerances Structure	166
Table 160:	Attitude File. Data Block	168
Table 161:	Attitude File. Attitude Angles Data	168
Table 163:	Attitude File. List of Attitude Angles	169
Table 164:	Attitude File. List of Quaternions Data	169
Table 165:	Attitude File. Attitude_Angles.....	169
Table 162:	Attitude File. Quaternions Data	169
Table 166:	Attitude File. Quaternions.....	170
Table 167:	Star Tracker Configuration File. Data Block	172
Table 168:	Star Tracker Configuration File. Mispointing.....	172
Table 169:	Star Tracker Configuration File. Star tracker limits.....	172
Table 170:	Star Tracker Configuration File. Star_Trackers_Priority.....	173
Table 171:	Star Tracker Configuration File. List_of_Star_Trackers	173
Table 172:	Star Tracker Configuration File. Pre and Post Launch angles	173
Table 173:	Star Tracker Configuration File. Rotation_Angles	173
Table 174:	DEM Configuration File. Data Block.....	176
Table 175:	DEM Configuration File. ACE Model	176
Table 176:	Swath Definition File. Data Block.....	179
Table 177:	Swath Definition File. Swath.....	179
Table 178:	Swath Definition File. Refraction	180
Table 179:	Swath Definition File. Swath Point	180
Table 180:	Swath Definition File. Point Geometry	180
Table 181:	Swath Definition File. Limb and Inertial Geometry	180
Table 182:	Swath Definition File. Distance Point Geometry	181
Table 183:	Swath Definition File. ASAR Geometry	181
Table 184:	Swath Definition File. Satellite Nominal Attitude.....	181
Table 185:	Swath Definition File. Satellite and Instrument Attitude	182
Table 186:	Swath Definition File. Parameter Model	182
Table 187:	Swath Definition File. Harmonic Model	182
Table 188:	Swath Definition File. File Model	183
Table 189:	Swath Definition File. Angle Model.....	183
Table 190:	Swath Definition File. Matrix Model	183
Table 191:	Swath Definition File. List_of_Parameters	183
Table 192:	Swath Definition File. List_of_Harmonics_Pitch/Roll/Yaw	183
Table 193:	Swath Definition File. Harmonic.....	184
Table 194:	Swath Definition File. Offsets	184
Table 195:	Swath Definition File. File.....	184
Table 196:	Swath Definition File. Time Selection.....	184
Table 197:	Swath Definition File. Time_Window	184

Table 198:	Swath Definition File. Row	184
Table 199:	Swath Template File. Variable_Header.....	187
Table 200:	Swath Template File. STF_Altitude	187
Table 201:	Swath Template File. Orbit_Geometry	187
Table 202:	Swath Template File. Orbit_State_Vector.....	188
Table 203:	Swath Template File. Line_Altitude	188
Table 204:	Swath Template File. Refraction.....	188
Table 205:	Swath Template File. Data_Block	189
Table 206:	Swath Template File. STF_Pt	189
Table 207:	Swath Template File. Geodetic_Pt.....	189
Table 208:	Swath Template File. Inertial_Pt.....	189
Table 209:	Zone Database File. Data_Block	192
Table 210:	Zone Database File. Zone	192
Table 211:	Zone Database File. Polygon_Pt.....	192
Table 212:	Station Database File. Data Block	195
Table 213:	Station Database File. Location	195
Table 214:	Station Database File. Mask_Point	195
Table 215:	Station Database File. Spacecraft	196
Table 216:	Known problems	200

1 SCOPE

The EXPLORER_DATA_HANDLING Software User Manual provides:

- a detailed description of usage of the CFI functions included within the EXPLORER_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 AND NOMENCLATURE

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

<i>CFI</i>	A group of CFI functions, and related software and documentation. that will be distributed by ESA to the users as an independent unit
<i>CFI function</i>	A single function within a CFI that can be called by the user
<i>Library</i>	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)

3 APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable Documents

[GEN_SUM]	Earth Explorer Mission CFI Software. General Software User Manual. EE-MA-DMS-GS-0002. Issue 3.7.3 07/05/10
[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 Explorer Mission CFI Software. Conventions Document. EE-MA-DMS-GS-0001. Issue 1.0. 27/10/09.
[MSC]	Earth Explorer Mission CFI Software. Mission Specific Customizations. EE-MA-DMS-GS-0018. Issue 1.0. 27/10/09.
[MSC]	Earth Explorer Mission CFI Software. Mission Specific Customizations. EE-MA-DMS-GS-0018. Issue 1.0. 27/10/09.
[F_H_SUM]	Earth Explorer Mission CFI Software. EXPLORER_FILE_HANDLING Software User Manual. EE-MA-DMS-GS-0008. Issue 3.7.3 07/05/10.
[LIB_SUM]	Earth Explorer Mission CFI Software. EXPLORER_LIB Software User Manual. EE-MA-DMS-GS-0003. Issue 3.7.3 07/05/10.
[ORBIT_SUM]	Earth Explorer Mission CFI Software. EXPLORER_ORBIT Software User Manual. EE-MA-DMS-GS-0004. Issue 3.7.3. 07/05/10.
[POINT_SUM]	Earth Explorer Mission CFI Software. EXPLORER_POINTING Software User Manual. EE-MA-DMS-GS-0005. Issue 3.7.3. 07/05/10.
[VISIB_SUM]	Earth Explorer Mission CFI Software. EXPLORER_VISIBILITY Software User Manual. EE-MA-DMS-GS-0006. Issue 3.7.3. 07/05/10.
[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 Explorer Mission Files. The following CFI functions are included:

4.1.1 Reading routines

- **xd_read_fhr:** reads the fixed header for an Earth Explorer XML file.
- **xd_read_bulletin:** reads the time correlations from an IERS bulletin.
- **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 a star tracker file for CRYOSAT.
- **xd_read_star_tracker_conf_file:** reads a 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

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**: frees 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.

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 EXPLORER_DATA_HANDLING software library, the following other CFI software libraries are required:

- EXPLORER_FILE_HANDLING (See [F_H_SUM]).

It is also needed to have properly installed in the system the following external GPL library:

- LIBXML2 (see [GEN_SUM]).

and the POSIX thread library:

- libpthread.so (pthread.lib for WINDOWS)

To use the EXPLORER_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):

- SOLARIS/LINUX:

```
-Icfi_include_dir -Lcfi_lib_dir -lexplorer_data_handling
                    -lexplorer_file_handling -lxml2 -lpthread
```

- WINDOWS:

```
/I "cfi_include_dir" /libpath:"cfi_lib_dir"
                                libexplorer_data_handling.lib
                                libexplorer_file_handling.lib
                                libxml2.lib pthread.lib
```

- MacOS:

```
-Icfi_include_dir -Lcfi_lib_dir -lexplorer_data_handling
                    -lexplorer_file_handling -lpthread
                    -framework libxml -framework libiconv
```

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

To avoid problems in linking a user application with the EXPLORER_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 necessary the xerces-c 2.7 (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 and the xerces-c dynamic library has to be included in the environment variable `PATH` (For LINUX and SOLARIS, the xerces library has to be included in the `LD_LIBRARY_PATH` instead).

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

Table 1: CFI functions included within EXPLORER_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

Function Name	Enumeration value	Long
xd_xml_validate	XD_XML_VALIDATE_ID	28
xd_read_tle	XD_READ_TLE	29
xd_write_tle	XD_WRITE_TLE	30
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 EXPLORER_DATA_HANDLING routines, as shown in the table below. The enumerations presented in [GEN_SUM] are also applicable.

Table 2: Enumerations within EXPLORER_DATA_HANDLING library

Input	Description	Enumeration value	Long
Boolean values	False value	XD_FALSE	0
	True value	XD_TRUE	1

Table 2: Enumerations within EXPLORER_DATA_HANDLING library

Input	Description	Enumeration value	Long
Returned status code	Error	XD_ERR	-1
	Ok status	XD_OK	0
	Warnig	XD_WARN	1
Satellite ID	Default Satellite 0	XD_SAT_DEFAULT	0
	Default Satellite 1	XD_SAT_DEFAULT1	1
	Default Satellite 2	XD_SAT_DEFAULT2	2
	Default Satellite 3	XD_SAT_DEFAULT3	3
	Default Satellite 4	XD_SAT_DEFAULT4	4
	Default Satellite 5	XD_SAT_DEFAULT5	5
	Default Satellite 6	XD_SAT_DEFAULT6	6
	Default Satellite 7	XD_SAT_DEFAULT7	7
	Default Satellite 8	XD_SAT_DEFAULT8	8
	Default Satellite 9	XD_SAT_DEFAULT9	9
	ERS 1	XD_SAT_ERS1	11
	ERS 2	XD_SAT_ERS2	12
	EnviSat	XD_SAT_ENVISAT	21
	Metop 1	XD_SAT_METOP1	31
	Metop 2	XD_SAT_METOP2	32
	Metop 3	XD_SAT_METOP3	33
	CryoSat	XD_SAT_CRYOSAT	41
	ADM	XD_SAT_ADM	51
	GOCE	XD_SAT_GOCE	61
	SMOS	XD_SAT_SMOS	71
	Terrasar	XD_SAT_TERRASAR	81
	EarthCARE	XD_SAT_EARTHCARE	91
	Swarm-A	XD_SAT_SWARM_A	101
	Swarm-B	XD_SAT_SWARM_B	102
	Swarm-C	XD_SAT_SWARM_C	103
	Sentinel-1A	XD_SAT_SENTINEL_1A	110
	Sentinel-1B	XD_SAT_SENTINEL_1B	111
	Sentinel-2	XD_SAT_SENTINEL_2	112
	Sentinel-3	XD_SAT_SENTINEL_3	113
	Seosat	XD_SAT_SEOSAT	120
Generic Satellite	XD_SAT_GENERIC	200	

Table 2: Enumerations within EXPLORER_DATA_HANDLING library

Input	Description	Enumeration value	Long
Time initialization mode	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_GEODETTIC_SWATH_TYPE	0
		XD_INERTIAL_SWATH_TYPE	1

Table 2: Enumerations within EXPLORER_DATA_HANDLING library

Input	Description	Enumeration value	Long
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
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

Table 2: Enumerations within EXPLORER_DATA_HANDLING library

Input	Description	Enumeration value	Long
Coordinate systems	Barycentric Mean of 2000.0	XD_BAR_MEAN_2000	0
	Heliocentric Mean of 2000.0	XD_HEL_MEAN_2000	1
	Geocentric Mean of 2000.0	XD_GEO_MEAN_2000	2
	Mean of date	XD_MEAN_DATE	3
	True of date	XD_TRUE_DATE	4
	Earth Fixed	XD_EARTH_FIXED	5
	Barycentric Mean of 1950.0	XD_BAR_MEAN_1950	6
	Galactic	XD_GALACTIC	7
	Satellite relative actual reference	XD_SAT_ACT_REF	8
	Quasi-Mean of Date	XD_QUASI_MEAN_DATE	9
	Pseudo-True of Date	XD_PSE_TRUE_DATE	10
	Quasi-True of Date	XD_QUASI_TRUE_DATE	11
	Topocentric	XD_TOPOCENTRIC	12
	Satellite reference	XD_SAT_REF	13
Satellite relative reference	XD_SAT_REL_REF	14	
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	XD_DEM_ACE_MODEL	0
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

Table 2: Enumerations within EXPLORER_DATA_HANDLING library

Input	Description	Enumeration value	Long
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

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 EXPLORER_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: EXPLORER_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
		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
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 UT1 and TAI
xd_iers_bulletin_b	Data for time correlations read from a IERS bulletin	table1	xd_bulb_table[100]	First table data in the IERS bulletin
		table2	xd_bulb_table[100]	Second table data in the IERS bulletin. If IERS bulletin is v2010, this table is filled with 0.
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

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
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
		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 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

Table 3: EXPLORER_DATA_HANDLING Structures

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			

Table 3: EXPLORER_DATA_HANDLING Structures

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_longitude	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	

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
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
		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		
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

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
xd_harmonic_data		num_terms	long[3]	Number of harmonics coefficient(pitch, roll and yaw)
		harmonic_type_pitch	long[XD_MAX_NUM_HARMONIC]	Harmonic type
		harmonic_type_roll	long[XD_MAX_NUM_HARMONIC]	Harmonic type
		harmonic_type_yaw	long[XD_MAX_NUM_HARMONIC]	Harmonic type
		harmonic_coef_pitch	double [XD_MAX_NUM_HARMONIC]	Harmonic coefficient
		harmonic_coef_roll	double [XD_MAX_NUM_HARMONIC]	Harmonic coefficient
		harmonic_coef_yaw	double [XD_MAX_NUM_HARMONIC]	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_model_str		angle_type	long	Angle type
		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_attitude_model_str	Attitude model structure	attitude_model	long	Attitude model type
		data	Attitude union data	Attitude union. One of the attitude structures.

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
Attitude union data	One of the following attitude structures	AOCS	long	AOCS model
		param_mode	xd_param_model_str	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_right	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
		swath_geom	xd_swath_geometry *	Swath geometry
		asar_geom	xd_asar_geometry	ASAR parameters
		sat_nom_att	xd_attitude_model_str *	Attitude data for sat. nominal att
		sat_att	xd_attitude_model_str *	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

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
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_LENGTH]	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
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

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
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 consecutive 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 consecutive quaternions
xd_tracker_conf_file	star trackers configuration 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 records

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
xd_dem_ace	DEM configuration data for ACE model	dir	char[100]	Directory where the DEM files are stored
		version	long	Getasse version
		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).
		y_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_config_file	DEM configuration data	model	long	DEM model
		dem_data	xd_dem_ace *	DEM ACE data
xd_dem_point	DEM file point	lon	double	longitude
		lat	double	latitude
		alt	double	altitude

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
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 (au/century)
		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
		antenna	char [XD_MAX_STR]	Describes the frequency band in which the antenna 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

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
		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		
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

Table 3: EXPLORER_DATA_HANDLING Structures

Structure name	Description	Structure Data		
		Variable Name	C type	Description
xd_zone_rec	Zone record data	zone_id	char [XD_MAX_STR]	Zone ID
		description	char [XD_MAX_STR]	Description 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)
xd_zone_file	Zone file structure	num_rec	long	Number of zones
		zone_rec	xd_zone_rec *	Array of zone records
xd_scf_appear	Appearance data for swath configuration files	colour	long	Colour (hexadecimal value from 0x000000 to 0xFFFFFFFF)
		draw	long	Draw (see enumeration in table 2)
		fill	long	Fill (see enumeration in table 2)
		opacity	long	Opacity (0-100%)

Table 3: EXPLORER_DATA_HANDLING Structures

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

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 Explorer 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: <ul style="list-style-type: none"> • = 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 EXPLORER_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 EXPLORER_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

7.1.6 Runtime performances

The following runtime performances have been measured.

Table 7: Runtime performances of xd_read_fhr function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
2.1	1.0	1.8	0.4

7.2 xd_read_bulletin

7.2.1 Overview

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

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 8: 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 9: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 9: Output parameters of `xd_read_bulletin` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 10: 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

7.2.6 Runtime performances

The following runtime performances have been measured.

Table 11: Runtime performances of `xd_read_bulletin` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
21.8	6.3	16.1	2.8

7.3 xd_read_orbit_file

7.3.1 Overview

The **xd_read_orbit_file** CFI function reads orbit files for Earth Explorer Missions. The files have to be written in XML and consists 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.

7.3.2 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);
}
```

7.3.3 Input parameters

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

Table 12: 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: <ul style="list-style-type: none"> a predicted orbit file a restituted orbit file or a DORIS Preliminary file 		<ul style="list-style-type: none"> XD_TRUE for ROF and DORIS files XD_FALSE for POF files

Table 12: Input parameters of `xd_read_orbit_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>time_orbit_flag</code>	<code>long*</code>	-	Flag for selecting the time range of the initialisation. Select either: <ul style="list-style-type: none"> • <code>XD_SEL_FILE</code>: for reading the whole file • <code>XD_SEL_ORBIT</code>: for reading the interval given by the <code>start_range</code> and the <code>stop_range</code> parameters in orbits • <code>XD_SEL_TIME</code>: for reading the interval given by the <code>start_range</code> and the <code>stop_range</code> parameters in days 	-	All
<code>time_ref</code>	<code>long*</code>	-	Time reference if <code>time_orbit_flag</code> is <code>XD_SEL_TIME</code> . Dummy otherwise.	-	-
<code>reading_osv_flag</code>	<code>long*</code>	-	flag to indicate if the state vectors data have to be read.	-	<ul style="list-style-type: none"> • <code>XD_TRUE</code> for reading the state vector data • <code>XD_FALSE</code> for reading just the times and orbit numbers
<code>start_range</code>	<code>double*</code>	-	Start orbit or day	orbits or days	-
<code>stop_range</code>	<code>double*</code>	-	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.

7.3.4 Output parameters

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

Table 13: Output parameters of `xd_read_orbit_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_orbit_file</code>	<code>long</code>	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 13: Output parameters of `xd_read_orbit_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>orbit_data</code>	<code>xd_orbit_file</code>	-	Data structure containing the data read from the file	-	-
<code>ierr</code>	<code>long[]</code>	-	Error vector	-	-

Memory Management: The `orbit_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_orbit_file`

7.3.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 14: 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	<code>XD_CFI_READ_ORBIT_FILE_READ_ERR</code>	0
ERR	Error in getting the first element inside the input range	No calculation performed	<code>XD_CFI_READ_ORBIT_FILE_INPUT_RANGE_ERR</code>	1
ERR	Error allocating memory	No calculation performed	<code>XD_CFI_READ_ORBIT_FILE_MEMORY_ERR</code>	2
ERR	Internal Error # 1	No calculation performed	<code>XD_CFI_READ_ORBIT_FILE_INTERNAL_1_ERR</code>	3
ERR	Error while reading data	No calculation performed	<code>XD_CFI_READ_ORBIT_FILE_DATA_READ_ERR</code>	4
ERR	Gap found after OSV no. %li	No calculation performed	<code>XD_CFI_READ_ORBIT_FILE_GAP_ERR</code>	5

7.3.6 Runtime performances

The following runtime performances have been measured.

Table 15: Runtime performances of xd_read_orbit_file function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
3.8	1.6	3.0	0.5

7.4 xd_free_orbit_file

7.4.1 Overview

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

7.4.2 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);
}
```

7.4.3 Input parameters

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

Table 16: 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	-	-

7.4.4 Output parameters

This function does not return any value nor parameters.

7.5 xd_read_doris

7.5.1 Overview

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

7.5.2 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);
}
```

7.5.3 Input parameters

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

Table 17: 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	-	<ul style="list-style-type: none"> • XD_SEL_FILE or • 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	-	<ul style="list-style-type: none"> • 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].

7.5.4 Output parameters

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

Table 18: Output parameters of `xd_read_doris` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_doris</code>	<code>long</code>	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>doris_data</code>	<code>xd_doris_file</code>	-	DORIS data	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.5.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 19: 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	<code>XD_CFI_READ_DORIS_ERROR_IN_MEMORY_ASIG_ERR</code>	0
ERR	Wrong input parameter value: <code>"time_mode"</code>	No calculation performed	<code>XD_CFI_READ_DORIS_WRONG_TIME_MODE_ERR</code>	1
ERR	Wrong time on input (start time after stop time)	No calculation performed	<code>XD_CFI_READ_DORIS_WRONG_TIME_1_ERR</code>	2

Table 19: Error messages of xd_read_doris function

Error type	Error message	Cause and impact	Error code	Error No
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_FI_NDKW_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

7.5.6 Runtime performances

The following runtime performances have been measured.

Table 20: Runtime performances of xd_read_doris function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
27.3	7.1	13.7	2.2

7.6 xd_free_doris

7.6.1 Overview

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

7.6.2 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 (&udoris_data);
}
```

7.6.3 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	-	-

7.6.4 Output parameters

This function does not return any value nor parameters.

7.7 xd_read_doris_header

7.7.1 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.

7.7.2 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);
}
```

7.7.3 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].

7.7.4 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
<code>xd_read_doris_header</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>doris_data</code>	<code>xd_doris_mph_sph</code>	-	doris header structure	-	-
<code>ierr</code>	long []	.	Error vector	-	-

7.7.5 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 EXPLORER_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 EXPLORER_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	<code>XD_CFI_READ_DORIS_HEADER_NO_FILENAME_ERR</code>	0
ERR	DORIS Level 0 file cannot be open	No calculation performed	<code>XD_CFI_READ_DORIS_HEADER_CANNOT_OPEN_ERR</code>	1
ERR	Could not find keyword: %s	No calculation performed	<code>XD_CFI_READ_DORIS_HEADER_FINDKW_ERROR_ERR</code>	2
ERR	Error reading DORIS data for keyword: %s	No calculation performed	<code>XD_CFI_READ_DORIS_HEADER_READ_ERR</code>	3

7.7.6 Runtime performances

The following runtime performances have been measured.

Table 25: Runtime performances of xd_read_doris_header function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
3.5	0.8	1.4	0.4

7.8 xd_read_osf

7.8.1 Overview

The `xd_read_osf` CFI function reads Orbit Scenario files for Earth Explorer Missions. The files have to be written in XML and consists 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.

7.8.2 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);
}
```

7.8.3 Input parameters

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

Table 26: 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	-	-

7.8.4 Output parameters

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

Table 27: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 27: Output parameters of `xd_read_osf` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>osf_data</code>	<code>xd_osf_file</code>	-	Structure with the OSF data	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.8.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 28: 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	<code>XD_CFI_READ_XML_OSF_INIT_PARSER_ERR</code>	0
ERR	Error finding the data block keyword	No calculation performed	<code>XD_CFI_READ_XML_OSF_XML_DATA_BLOCK_ERR</code>	1
ERR	Error reading the data block attribute	No calculation performed	<code>XD_CFI_READ_XML_OSF_XML_ATTRIBUTE_ERR</code>	2
ERR	"Error reading the xml attribute"	No calculation performed	<code>XD_CFI_READ_XML_OSF_XML_TYPE_ERR</code>	3
ERR	Error reading XML element: %s	No calculation performed	<code>XD_CFI_READ_XML_OSF_READ_PARAM_ERR</code>	4
ERR	Error the size of the list (negative)	No calculation performed	<code>XD_CFI_READ_XML_OSF_XML_DATA_BLOCK_SIZE_ERR</code>	5
ERR	Error allocating memory	No calculation performed	<code>XD_CFI_READ_XML_OSF_MEMORY_ERR</code>	6

7.8.6 Runtime performances

The following runtime performances have been measured.

Table 29: Runtime performances of xd_read_osf function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
10.8	2.5	4.2	0.7

7.9 xd_free_osf

7.9.1 Overview

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

7.9.2 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);
}
```

7.9.3 Input parameters

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

Table 30: 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	-	-

7.9.4 Output parameters

This function does not return any value nor parameters.

7.10 xd_read_sdf

7.10.1 Overview

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

7.10.2 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);
}
```

7.10.3 Input parameters

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

Table 31: 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	-	-

7.10.4 Output parameters

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

Table 32: Output parameters of `xd_read_sdf` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_sdf</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>sdf_data</code>	<code>xd_sdf_file</code>	-	Swath Definition data structure	-	-
<code>ierr</code>	<code>long[]</code>	-	Error vector	-	-

Memory Management: The `sdf_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_sdf`.

7.10.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 33: 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	<code>XD_CFI_READ_SDF_OPEN_FILE_ERR</code>	0
ERR	Error allocating memory	No calculation performed	<code>XD_CFI_READ_SDF_MEMORY_ERR</code>	1
ERR	Error reading swath record %d	No calculation performed	<code>XD_CFI_READ_SDF_RECORD_READ_ERR</code>	2
ERR	Could not get file version	No calculation performed	<code>XD_CFI_READ_SDF_VERSION_ERR</code>	3

7.10.6 Runtime performances

The following runtime performances have been measured.

Table 34: Runtime performances of xd_read_sdf function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
1.6	0.8	1.7	0.3

7.11 xd_free_sdf

7.11.1 Overview

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

7.11.2 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);
}
```

7.11.3 Input parameters

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

Table 35: 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	-	-

7.11.4 Output parameters

This function does not return any value nor parameters.

7.12 xd_read_stf

7.12.1 Overview

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

7.12.2 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);
}
```

7.12.3 Input parameters

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

Table 36: 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	-	-

7.12.4 Output parameters

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

Table 37: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 37: Output parameters of `xd_read_stf` function

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

7.12.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 38: 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	egative 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_RECORD_READ_ERR	8

Table 38: Error messages of *xd_read_stf* function

Error type	Error message	Cause and impact	Error code	Error No
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

7.12.6 Runtime performances

The following runtime performances have been measured.

Table 39: Runtime performances of *xd_read_stf* function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
915.1	395.8	384.5	66.6

7.13 xd_free_stf

7.13.1 Overview

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

7.13.2 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);
}
```

7.13.3 Input parameters

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

Table 40: 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	-	-

7.13.4 Output parameters

This function does not return any value nor parameters.

7.14 xd_read_stf_vhr

7.14.1 Overview

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

7.14.2 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);
}
```

7.14.3 Input parameters

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

Table 41: Input parameters of `xd_read_stf_vhr` function

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

7.14.4 Output parameters

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

Table 42: Output parameters of `xd_read_stf_vhr` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_stf_vhr</code>	<code>long</code>	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>vhr_data</code>	<code>xd_stf_vhr</code>	-	Data structure for the Swath template variable header	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.14.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 43: 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

7.14.6 Runtime performances

The following runtime performances have been measured.

Table 44: Runtime performances of xd_read_stf_vhr function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
437.0	186.3	173.8	26.3

7.15 xd_free_stf_vhr

7.15.1 Overview

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

7.15.2 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);
}
```

7.15.3 Input parameters

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

Table 45: 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	-	-

7.15.4 Output parameters

This function does not return any value nor parameters.

7.16 xd_read_att

7.16.1 Overview

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

7.16.2 Calling interface

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

```
#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);
}
```

7.16.3 Input parameters

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

Table 46: 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	-	-

7.16.4 Output parameters

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

Table 47: Output parameters of xd_read_att function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xd_read_att	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 47: Output parameters of `xd_read_att` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>att_data</code>	<code>xd_att_file</code>	-	Attitude data structure	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.16.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 48: 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	<code>XD_CFI_READ_ATT_INIT_PARSER_ERR</code>	0
ERR	Error reading element: %s	No calculation performed	<code>XD_CFI_READ_ATT_READ_PARAM_ERR</code>	1
ERR	Wrong file type	No calculation performed	<code>XD_CFI_READ_ATT_WRONG_FILE_TYPE_ERR</code>	2
ERR	Error navigating through the file	No calculation performed	<code>XD_CFI_READ_XML_ATT_NAVIGATION_ERR</code>	3
ERR	Wrong attitude data type. Only "Quaternions" and "Attitude_Angles_Data" allowed	No calculation performed	<code>XD_CFI_READ_ATT_WRONG_DATA_TYPE_ERR</code>	4
ERR	Inconsistent values for <Attitude_Data_Type> and the list of attitude data	No calculation performed	<code>XD_CFI_READ_ATT_INCONSISTENT_DATA_TYPE_ERR</code>	5
ERR	Wrong number of records in the list	No calculation performed	<code>XD_CFI_READ_ATT_XML_DATA_BLOCK_SIZE_ERR</code>	6
ERR	Wrong parameter in "Inertial_Ref_Frame"	No calculation performed	<code>XD_CFI_READ_ATT_WRONG_REF_FRAME_ERR</code>	7

Table 48: Error messages of xd_read_att function

Error type	Error message	Cause and impact	Error code	Error No
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

7.16.6 Runtime performances

The following runtime performances have been measured.

Table 49: Runtime performances of xd_read_att function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
2.4	1.0	2.0	0.4

7.17 xd_free_att

7.17.1 Overview

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

7.17.2 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);
}
```

7.17.3 Input parameters

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

Table 50: 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	-	-

7.17.4 Output parameters

This function does not return any value nor parameters.

7.18 xd_read_star_tracker

7.18.1 Overview

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

7.18.2 Calling interface

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

```
#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);
}
```

7.18.3 Input parameters

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

Table 51: 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		<ul style="list-style-type: none"> • 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)

Table 51: Input parameters of `xd_read_star_tracker` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>str_limit</code>	<code>xd_str_limits</code>	-	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.

7.18.4 Output parameters

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

Table 52: Output parameters of `xd_read_star_tracker` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_star_tracker</code>	<code>long</code>	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>str_data</code>	<code>xd_star_tracker_file</code>	-	Star tracker data structure	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.18.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 53: 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

7.18.6 Runtime performances

The following runtime performances have been measured.

Table 54: Runtime performances of *xd_read_star_tracker* function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
134.9	32.0	63.8	7.9

7.19 xd_free_star_tracker

7.19.1 Overview

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

7.19.2 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 (&u>str_data);
}
```

7.19.3 Input parameters

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

Table 55: 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	-	-

7.19.4 Output parameters

This function does not return any value nor parameters.

7.20 xd_read_star_tracker_conf_file

7.20.1 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.

7.20.2 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);
}
```


7.20.3 Input parameters

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

Table 56: Input parameters of `xd_read_star_tracker_conf_file` function

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

7.20.4 Output parameters

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

Table 57: Output parameters of `xd_read_star_tracker_conf_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_star_tracker_conf_file</code>	<code>long</code>	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>conf_data</code>	<code>xd_tracker_conf_file</code>	-	Star tracker configuration data structure with	-	-
<code>ierr</code>	<code>long[]</code>	-	Error vector	-	-

7.20.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM]).

Table 58: 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

7.20.6 Runtime performances

The following runtime performances have been measured.

Table 59: Runtime performances of xd_read_star_tracker_conf_file function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
240.6	95.2	89.9	14.2

7.21 xd_read_dem

7.21.1 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.

7.21.2 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);
}
```

7.21.3 Input parameters

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

Table 60: 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.

7.21.4 Output parameters

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

Table 61: Output parameters of `xd_read_dem` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_dem</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>dem_data</code>	<code>xd_dem_file</code>	-	DEM data structure	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.21.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 62: 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	<code>XD_CFI_READ_DEM_MEMORY_ERR</code>	0
ERR	Incorrect input DEM configuration file	No calculation performed	<code>XD_CFI_READ_DEM_NO_CONFIG_FILE_ERR</code>	1
ERR	Wrong input file name	No calculation performed	<code>XD_CFI_READ_DEM_WRONG_FILENAME_ERR</code>	2
ERR	Could not open the DEM file	No calculation performed	<code>XD_CFI_READ_DEM_OPEN_FILE_ERR</code>	3
ERR	Could not read the DEM file	No calculation performed	<code>XD_CFI_READ_DEM_READ_FILE_ERR</code>	5

7.21.6 Runtime performances

The following runtime performances have been measured.

Table 63: Runtime performances of xd_read_dem function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
4344.0	1078.0	2211.0	423.0

7.22 xd_free_dem

7.22.1 Overview

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

7.22.2 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);
}
```

7.22.3 Input parameters

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

Table 64: 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	-	-

7.22.4 Output parameters

This function does not return any value nor parameters.

7.23 xd_read_dem_config_file

7.23.1 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).

7.23.2 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);
}
```

7.23.3 Input parameters

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

Table 65: 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	-	-

7.23.4 Output parameters

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

Table 66: Output parameters of `xd_read_dem_config_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_dem_config_file</code>	long	-	Function status flag: <ul style="list-style-type: none"> = 0 No error > 0 Warnings, results generated < 0 Error, no results generated 	-	-
<code>dem_config_data</code>	<code>xd_dem_config_file</code>	-	DEM configuration data structure	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.23.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 67: 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 configuration file	No calculation performed	<code>XD_CFI_READ_DEM_CONFIG_FILE_OPEN_ERR</code>	0
ERR	Could not read the configuration file	No calculation performed	<code>XD_CFI_READ_DEM_CONFIG_FILE_READ_ERR</code>	1
ERR	Could not open the model tag	No calculation performed	<code>XD_CFI_READ_DEM_CONFIG_FILE_READ_MODEL_ERR</code>	2
ERR	Memory allocation error	No calculation performed	<code>XD_CFI_READ_DEM_CONFIG_FILE_MEMORY_ERR</code>	3
ERR	Could not open a ACE file	No calculation performed	<code>XD_CFI_READ_DEM_CONFIG_FILE_OPEN_DEM_FILE_ERR</code>	4
ERR	Could not read a ACE file	No calculation performed	<code>XD_CFI_READ_DEM_CONFIG_FILE_READ_DEM_FILE_ERR</code>	5



Code: EE-MA-DMS-GS-0007
Date: 07/05/2010
Issue: 3.7.3
Page: 97

7.24 xd_read_zone

7.24.1 Overview

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

7.24.2 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);
}
```

7.24.3 Input parameters

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

Table 68: 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	-	-

7.24.4 Output parameters

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

Table 69: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 69: Output parameters of `xd_read_zone` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
zone_rec	xd_zone_rec	-	Zone Data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The `zone_rec` 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`.

7.24.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 70: 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

Table 70: Error messages of xd_read_zone function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error reading zone record	No calculation performed	XD_CFI_READ_ZONE_RECORD_READ_ERR	9

7.24.6 Runtime performances

The following runtime performances have been measured.

Table 71: Runtime performances of xd_read_zone function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
112.2	48.3	45.5	6.8

7.25 xd_free_zone

7.25.1 Overview

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

7.25.2 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 (&u>zone_data);
}
```

7.25.3 Input parameters

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

Table 72: 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	-	-

7.25.4 Output parameters

This function does not return any value nor parameters.

7.26 xd_read_zone_file

7.26.1 Overview

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

7.26.2 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);
}
```

7.26.3 Input parameters

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

Table 73: 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	-	-

7.26.4 Output parameters

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

Table 74: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 74: Output parameters of `xd_read_zone_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_zone_file</code>	<code>zone_data</code>	-	Structure containing the data for all the zones read from the file	-	-
<code>ierr</code>	<code>long[]</code>	-	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`.

7.26.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 75: 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	<code>XD_CFI_READ_ZONE_FILE_INIT_PARSER_ERR</code>	0
ERR	Data Block not found	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_DATA_BLOCK_ERR</code>	1
ERR	Data Block attribute not read.	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_DATA_BLOCK_ATTRIBUTE_ERR</code>	2
ERR	Data Block not of XML type.	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_XML_TYPE_ERR</code>	3
ERR	List_of_Zones not found.	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_LIST_ZONES_READ_ERR</code>	4
ERR	List_of_Zones attribute not read	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_LIST_ZONES_SIZE_ERR</code>	5
ERR	Error allocating memory	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_MEM_ERR</code>	6
ERR	Error reading zone record number %d	No calculation performed	<code>XD_CFI_READ_ZONE_FILE_RECORD_READ_ERR</code>	7

7.26.6 Runtime performances

The following runtime performances have been measured.

Table 76: Runtime performances of xd_read_zone_file function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
121.9	53.1	53.6	10.2

7.27 xd_free_zone_file

7.27.1 Overview

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

7.27.2 Calling interface

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

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

7.27.3 Input parameters

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

Table 77: 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	-	-

7.27.4 Output parameters

This function does not return any value nor parameters.

7.28 xd_read_zone_id

7.28.1 Overview

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

7.28.2 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);
}
```

7.28.3 Input parameters

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

Table 78: 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	-	-

7.28.4 Output parameters

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

Table 79: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 79: Output parameters of `xd_read_zone_id` function

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

7.28.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 80: 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_I NIT_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

7.28.6 Runtime performances

The following runtime performances have been measured.

Table 81: Runtime performances of xd_read_zone_id function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
116.3	49.7	47.1	7.4

7.29 xd_free_zone_id

7.29.1 Overview

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

7.29.2 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 (&u>zone_ids);
}
```

7.29.3 Input parameters

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

Table 82: 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	-	-

7.29.4 Output parameters

This function does not return any value nor parameters.

7.30 xd_read_station

7.30.1 Overview

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

7.30.2 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);
}
```

7.30.3 Input parameters

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

Table 83: 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)	-	-

7.30.4 Output parameters

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

Table 84: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 84: Output parameters of `xd_read_station` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
station_rec	xd_station_rec	-	Station record data	-	-
ierr	long[]	-	Error vector	-	-

7.30.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 85: 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_I NIT_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_I NTERNAL_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

7.30.6 Runtime performances

The following runtime performances have been measured.

Table 86: Runtime performances of xd_read_station function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
155.9	66.7	62.1	9.3

7.31 xd_read_station_file

7.31.1 Overview

The `xd_read_station_file` CFI function reads a whole station file for Earth Explorer Missions.

7.31.2 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);
}
```

7.31.3 Input parameters

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

Table 87: 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	-	-

7.31.4 Output parameters

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

Table 88: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 88: Output parameters of `xd_read_station_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
station_data	xd_station_file	-	Station file data structure	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The `station_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_station_file`.

7.31.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 89: 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_ERR	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

7.31.6 Runtime performances

The following runtime performances have been measured.

Table 90: Runtime performances of `xd_read_station_file` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
177.0	75.0	81.0	13.0

7.32 xd_free_station_file

7.32.1 Overview

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

7.32.2 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 (&u>station_data);
}
```

7.32.3 Input parameters

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

Table 91: 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	-	-

7.32.4 Output parameters

This function does not return any value nor parameters.

7.33 xd_read_station_id

7.33.1 Overview

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

7.33.2 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);
}
```

7.33.3 Input parameters

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

Table 92: 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	-	-

7.33.4 Output parameters

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

Table 93: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
num_stations	long	-	Number of stations	-	-

Table 93: Output parameters of `xd_read_station_id` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
station_list	char**	.	Station list name	-	-
ierr	long[]	-	Error vector	-	-

Memory Management: The `station_list` is a double pointer to memory allocated dynamically. 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`.

7.33.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 94: 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

7.33.6 Runtime performances

The following runtime performances have been measured.

Table 95: Runtime performances of xd_read_station_id function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
167.2	69.1	67.9	10.1

7.34 xd_free_station_id

7.34.1 Overview

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

7.34.2 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 (&u>station_ids);
}
```

7.34.3 Input parameters

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

Table 96: 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	-	-

7.34.4 Output parameters

This function does not return any value nor parameters.

7.35 xd_read_star

7.35.1 Overview

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

7.35.2 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);
}
```

7.35.3 Input parameters

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

Table 97: 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	-	-

7.35.4 Output parameters

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

Table 98: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
star_data	xd_star_rec	-	Star data structure	-	-

Table 98: Output parameters of `xd_read_star` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>ierr</code>	<code>long[]</code>	-	Error vector	-	-

7.35.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 99: 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	<code>XD_CFI_READ_STAR_FILE_NOT_FOUND_ERR</code>	0
ERR	star id. %s not found in the star database file	No calculation performed	<code>XD_CFI_READ_STAR_STAR_NOT_FOUND_ERR</code>	1

7.35.6 Runtime performances

The following runtime performances have been measured.

Table 100: Runtime performances of `xd_read_star` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
18.1	4.5	6.3	1.9

7.36 xd_read_star_file

7.36.1 Overview

The `xd_read_star_file` CFI function reads a star database file for Earth Explorer Missions.

7.36.2 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);
}
```

7.36.3 Input parameters

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

Table 101: 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)	-	-

7.36.4 Output parameters

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

Table 102: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
star_data	xd_star_file	-	Star file structure	-	-

Table 102: Output parameters of `xd_read_star_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr	long[]	-	Error vector	-	-

Memory Management: The `star_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_file`.

7.36.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 103: 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_ERR	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

7.36.6 Runtime performances

The following runtime performances have been measured.

Table 104: Runtime performances of `xd_read_star_file` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
1555	361	496	156

7.37 xd_read_star_id

7.37.1 Overview

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

7.37.2 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);
}
```

7.37.3 Input parameters

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

Table 105: 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	-	-

7.37.4 Output parameters

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

Table 106: 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: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-

Table 106: Output parameters of `xd_read_star_id` function

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

7.37.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 107: 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 data-base 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

7.37.6 Runtime performances

The following runtime performances have been measured.

Table 108: Runtime performances of `xd_read_star_id` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
1270	291	491	65

7.38 xd_read_tle

7.38.1 Overview

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

7.38.2 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);
}
```

7.38.3 Input parameters

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

Table 109: Input parameters of `xd_read_tle` function

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

7.38.4 Output parameters

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

Table 110: Output parameters of `xd_read_tle` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_read_tle</code>	<code>long</code>	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>tle_data</code>	<code>xd_tle_file</code>	-	Orbital state vectors data structure	-	-
<code>ierr</code>	<code>long[]</code>	-	Error vector	-	-

Memory Management: The `tle_data` is a pointer to memory allocated dynamically. 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`.

7.38.5 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 EXPLORER_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 calling the function of the EXPLORER_DATA_HANDLING software library **xd_get_code** (see [GEN_SUM]).

Table 111: 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

7.38.6 Runtime performances

The following runtime performances have been measured.

Table 112: Runtime performances of xd_read_tle function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
TBC	TBC	TBC	TBC

7.39 xd_free_tle

7.39.1 Overview

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

7.39.2 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 (&u>tle_data);
}
```

7.39.3 Input parameters

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

Table 113: 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	-	-

7.39.4 Output parameters

This function does not return any value nor parameters.

7.40 xd_write_orbit_file

7.40.1 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

7.40.2 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);
}
```

7.40.3 Input parameters

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

Table 114: 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	-	-

7.40.4 Output parameters

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

Table 115: Output parameters of `xd_write_orbit_file` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_write_orbit_file</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.40.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 116: 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	<code>XD_CFI_WRITE_ORBIT_FILE_CREATE_TREE_ERR</code>	0
ERR	Cannot create in-memory XML tree	No calculation performed	<code>XD_CFI_WRITE_ORBIT_FILE_CREATE_ROOT_ERR</code>	1
ERR	Cannot write the fixed header	No calculation performed	<code>XD_CFI_WRITE_ORBIT_FILE_WRITE_FHR_ERR</code>	2
ERR	Cannot add XML node to tree: %s	No calculation performed	<code>XD_CFI_WRITE_ORBIT_FILE_CREATE_NODE_ERR</code>	3
ERR	Cannot convert time from processing to external	No calculation performed	<code>XD_CFI_WRITE_ORBIT_FILE_GET_ASCII_TIME_ERR</code>	4
ERR	Cannot write XML file	No calculation performed	<code>XD_CFI_WRITE_ORBIT_FILE_WRITE_ERR</code>	5
ERR	Cannot go to the desired node	No calculation performed	<code>XD_CFI_WRITE_ORBIT_FILE_GOTO_NODE_ERR</code>	6
WARN	Cannot write schema in the file	File written to disk but without schema	<code>XD_CFI_WRITE_ORBIT_FILE_SET_SCHEMA_WARN</code>	7

7.40.6 Runtime performances

The following runtime performances have been measured.

Table 117: Runtime performances of xd_write_orbit_file function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
17.0	7.3	8.2	4.8

7.41 xd_write_osf

7.41.1 Overview

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

7.41.2 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);
}
```

7.41.3 Input parameters

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

Table 118: 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	-	-

7.41.4 Output parameters

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

Table 119: Output parameters of `xd_write_osf` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_write_osf</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.41.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 120: 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	<code>XD_CFI_WRITE_OSF_CREATE_TREE_ERR</code>	0
ERR	Cannot write the fixed header	No calculation performed	<code>XD_CFI_WRITE_OSF_WRITE_FHR_ERR</code>	1
ERR	Cannot create root element	No calculation performed	<code>XD_CFI_WRITE_OSF_CREATE_ROOT_ERR</code>	2
ERR	Cannot add XML node to tree	No calculation performed	<code>XD_CFI_WRITE_OSF_CREATE_NODE_ERR</code>	3
ERR	Cannot set XML node value	No calculation performed	<code>XD_CFI_WRITE_OSF_SET_NODE_VALUE_ERR</code>	4
ERR	Cannot convert time from processing to external	No calculation performed	<code>XD_CFI_WRITE_OSF_TIME_TO_EXTERNAL_ERR</code>	5
ERR	Cannot write XML file	No calculation performed	<code>XD_CFI_WRITE_OSF_WRITE_ERR</code>	6
WARN	Cannot write schema in the file	File written to disk but without schema	<code>XD_CFI_WRITE_OSF_SET_SCHEMA_WARN</code>	7

7.41.6 Runtime performances

The following runtime performances have been measured.

Table 121: Runtime performances of xd_write_osf function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
42.0	8.0	10.0	6.0

7.42 xd_write_doris

7.42.1 Overview

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

7.42.2 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);
}
```

7.42.3 Input parameters

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

Table 122: 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	-	-

7.42.4 Output parameters

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

Table 123: Output parameters of `xd_write_doris` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_write_doris</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.42.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 124: 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	<code>XD_CFI_WRITE_DORIS_OPEN_ERR</code>	0
ERR	Error writing the fixed header	No calculation performed	<code>XD_CFI_WRITE_DORIS_WRITE_FHR_ERR</code>	1
ERR	Error writing the binary data	No calculation performed	<code>XD_CFI_WRITE_DORIS_WRITE_BINARY_ERR</code>	2

7.42.6 Runtime performances

The following runtime performances have been measured.

Table 125: Runtime performances of `xd_write_doris` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
103.0	15.0	44.0	16.0

7.43 xd_write_stf

7.43.1 Overview

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

7.43.2 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);
}
```

7.43.3 Input parameters

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

Table 126: 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	-	-

7.43.4 Output parameters

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

Table 127: Output parameters of `xd_write_stf` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_write_stf</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.43.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM])

Table 128: 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	<code>XD_CFI_WRITE_STF_CREATE_XML_ERR</code>	0
ERR	Cannot create root node in the XML tree.	No calculation performed	<code>XD_CFI_WRITE_STF_CREATE_ROOT_XML_ERR</code>	1
ERR	Error writing fixed header.	No calculation performed	<code>XD_CFI_WRITE_STF_XD_FHR_WRITE_ERR</code>	2
ERR	Error while writing Swath Template File variable header.	No calculation performed	<code>XD_CFI_WRITE_STF_XD_STF_VHR_WRITE_ERR</code>	3
ERR	Cannot create the node %s	No calculation performed	<code>XD_CFI_WRITE_STF_CREATE_NODE_ERR</code>	4
ERR	Wrong swath_type	No calculation performed	<code>XD_CFI_WRITE_STF_WRONG_SWATH_TYPE_ERR</code>	5
ERR	Error while writing the swath record n.%d	No calculation performed	<code>XD_CFI_WRITE_STF_WRITE_REC_ERR</code>	6
ERR	Cannot write to disk the XML tree	No calculation performed	<code>XD_CFI_WRITE_STF_WRITE_ERR</code>	7
WARN	Cannot write schema in the file	File written to disk but without schema	<code>XD_CFI_WRITE_STF_SET_SCHEMA_WARN</code>	8

7.43.6 Runtime performances

The following runtime performances have been measured.

Table 129: Runtime performances of xd_write_stf function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
1639	652	713	66.6

7.44 xd_write_att

7.44.1 Overview

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

7.44.2 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);
}
```

7.44.3 Input parameters

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

Table 130: 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	-	-

7.44.4 Output parameters

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

Table 131: Output parameters of `xd_write_att` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_write_att</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.44.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 132: 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	<code>XD_CFI_WRITE_ATT_CREATE_TREE_ERR</code>	0
ERR	Cannot create root element	No calculation performed	<code>XD_CFI_WRITE_ATT_CREATE_ROOT_ERR</code>	1
ERR	Cannot write the fixed header	No calculation performed	<code>XD_CFI_WRITE_ATT_WRITE_FHR_ERR</code>	2
ERR	Cannot add XML node to tree: %s	No calculation performed	<code>XD_CFI_WRITE_ATT_CREATE_NODE_ERR</code>	3
ERR	Cannot convert time from processing to external	No calculation performed	<code>XD_CFI_WRITE_ATT_GET_ASCII_TIME_ERR</code>	4
ERR	Cannot go to the desired node	No calculation performed	<code>XD_CFI_WRITE_ATT_GOTO_NODE_ERR</code>	5
ERR	Cannot write XML file	No calculation performed	<code>XD_CFI_WRITE_ATT_WRITE_ERR</code>	6
WARN	Cannot write schema in the file	File written to disk but without schema	<code>XD_CFI_WRITE_ATT_SET_SCHEMA_WARN</code>	7

7.44.6 Runtime performances

The following runtime performances have been measured.

Table 133: Runtime performances of xd_write_att function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
25.0	7.0	8.0	5.0

7.45 xd_write_tle

7.45.1 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.

7.45.2 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);
}
```

7.45.3 Input parameters

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

Table 134: 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	-	-

7.45.4 Output parameters

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

Table 135: Output parameters of `xd_write_tle` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_write_tle</code>	long	-	Function status flag: <ul style="list-style-type: none"> • = 0 No error • > 0 Warnings, results generated • < 0 Error, no results generated 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.45.5 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 EXPLORER_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 EXPLORER_DATA_HANDLING software library `xd_get_code` (see [GEN_SUM]).

Table 136: 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	<code>XD_WRITE_TLE_FILE_OPEN_ERR</code>	0
ERR	Could not write the TLE file: %s	No calculation performed	<code>XD_WRITE_TLE_WRITE_ERR</code>	1

7.45.6 Runtime performances

The following runtime performances have been measured.

Table 137: Runtime performances of `xd_write_tle` function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
TBC	TBC	TBC	TBC

7.46 xd_xml_validate

7.46.1 Overview

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

7.46.2 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);
}
```

7.46.3 Input parameters

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

Table 138: 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: <ul style="list-style-type: none"> • 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 <i>schema</i> 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]	-	-

7.46.4 Output parameters

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

Table 139: Output parameters of `xd_xml_validate` function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
<code>xd_xml_validate</code>	long	-	Function status flag: <ul style="list-style-type: none"> = 0 No error > 0 Warnings, results generated < 0 Error, no results generated 	-	-
<code>valid_status</code>	long	-	The result of the validation: <ul style="list-style-type: none"> <code>XD_XML_INVALID</code> (= -1) <code>XD_XML_VALID</code> (= 0) 	-	-
<code>ierr</code>	long[]	-	Error vector	-	-

7.46.5 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 `EXPLORER_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 `EXPLORER_DATA_HANDLING` software library `xd_get_code` (see [GEN_SUM]).

Table 140: 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.	<code>XD_CFI_XML_VALIDATE_SCHEMA_ERR</code>	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.	<code>XD_CFI_XML_VALIDATE_INIT_PARSER_ERR</code>	1
ERR	Memory allocation error.	No calculation performed.	<code>XD_CFI_XML_VALIDATE_MEMORY_ERR</code>	2
ERR	No schema provided.	No calculation performed.	<code>XD_CFI_XML_VALIDATE_NO_SCHEMA_ERR</code>	3
ERR	Wrong input mode.	No calculation performed.	<code>XD_CFI_XML_VALIDATE_WRONG_MODE_ERR</code>	4

Table 140: Error messages of xd_xml_validate function

Error type	Error message	Cause and impact	Error code	Error No
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_INCONSISTENT_SCHEMA_VERSIONS_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_VERSION_WARN	11
WARN	Schema version in XML file is newer than the schema version	Calculation performed	XD_CFI_XML_VALIDATE_GREATER_SCHEMA_VERSION_WARN	12

7.46.6 Runtime performances

The following runtime performances have been measured.

Table 141: Runtime performances of xd_xml_validate function

Solaris 32-bit. [ms]	Solaris 64 bit. [ms]	Linux 32-bit. [ms]	Linux 64-bit. [ms]
388.2	253.4	181.89	93.4

7.46.7 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
```

8 LIST OF SCHEMA'S VERSIONS

Next table shows 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".

Table 142: List of schema's versions

File Type	Version	Supp. CFI's	Schema
Predicted Orbit files	1.0	N	EO_OPER_MPL_ORBPRES_0100.XSD
	1.1	Y	EO_OPER_MPL_ORBPRES_0101.XSD
	1.2	N	EO_OPER_MPL_ORBPRES_0102.XSD
Restituted Orbit files	1.0	N	EO_OPER_AUX_ORBRES_0100.XSD
	1.1	Y	EO_OPER_AUX_ORBRES_0101.XSD
	1.2	N	EO_OPER_AUX_ORBRES_0102.XSD
Doris Preliminary files	1.0	N	EO_OPER_AUX_ORBDOP_0100.XSD
	1.1	Y	EO_OPER_AUX_ORBDOP_0101.XSD
	1.2	N	EO_OPER_AUX_ORBDOP_0102.XSD
Doris Precise files	1.0	N	EO_OPER_AUX_ORBDOR_0100.XSD
	1.1	Y	EO_OPER_AUX_ORBDOR_0101.XSD
	1.2	N	EO_OPER_AUX_ORBDOR_0102.XSD
Orbit Scenario files	1.0	N	EO_OPER_MPL_ORBSCT_0100.XSD
	1.1	Y	EO_OPER_MPL_ORBSCT_0101.XSD
Orbit Event files	1.0	N	EO_OPER_MPL_ORBREF_0100.XSD
	1.1	Y	EO_OPER_MPL_ORBREF_0101.XSD
	1.2	N	EO_OPER_MPL_ORBREF_0102.XSD
DORIS Navigator files	1.1	Y	EO_OPER_DOR_NAV_0_.HeaderTypes_0101.XSD with EO_OPER_DOR_NAV_0_.DataBlockTypes_0100.XSD
Star Tracker files	1.1	R	EO_OPER_STR1ATT_0_HeaderTypes_0101.XSD with EO_OPER_STR1ATT_0_.DataBlockTypes_0100.XSD
Satellite Configuration File	1.0	N	EO_OPER_INT_SATCFG_0100.XSD
	1.1	N	EO_OPER_INT_SATCFG_0101.XSD
	1.2	R	EO_OPER_INT_SATCFG_0102.XSD

Table 142: List of schema's versions

Attitude File	1.0	N	EO_OPER_INT_ATTCFG_0100.XSD
	1.1	Y	EO_OPER_INT_ATTREF_0101.XSD
Star tracker configuration File	1.0	N	EO_OPER_INT_STRCFG_0100.XSD
	1.1	R	EO_OPER_INT_STRCFG_0101.XSD
DEM Configuration File	1.0	N	EO_OPER_INT_DEMCFG_0100.XSD
	1.1	R	EO_OPER_INT_DEMCFG_0101.XSD
	1.2	R	EO_OPER_INT_DEMCFG_0102.XSD
Swath Definition File	1.0	N	EO_OPER_MPL_SW_DEF_0100.XSD
	1.1	R	EO_OPER_MPL_SW_DEF_0101.XSD
	2.1	R	EO_OPER_MPL_SW_DEF_0201.XSD
Swath Template File	1.0	N	EO_OPER_MPL_SWTREF_0100.XSD
	1.1	R	EO_OPER_MPL_SWTREF_0101.XSD
	2.0	Y	EO_OPER_MPL_SWTREF_0200.XSD
Zone Database File	1.0	N	EO_OPER_MPL_ZON_DB_0100.XSD
	1.1	R	EO_OPER_MPL_ZON_DB_0101.XSD
Station Database File	1.0	N	EO_OPER_MPL_GND_DB_0100.XSD
	1.1	R	EO_OPER_MPL_GND_DB_0101.XSD
	1.2	R	EO_OPER_MPL_GND_DB_0102.XSD
	1.3	R	EO_OPER_MPL_GND_DB_0103.XSD
	1.4	R	EO_OPER_MPL_GND_DB_0104.XSD
Precise Propagator Configuration File	1.0	R	EO_OPER_INT_PPRCFG_0100.XSD

9 FILES FORMAT SPECIFICATION

This section presents the formats for all the files used by the Earth Explorer CFI software.

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 Explorer 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 Explorer 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 following tables:

Table 143: 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).

Table 143: Fixed Header Structure

Tag name	type	Attribute	C Format	Description
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 144)	-	-	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 145)	-	-	Structure with information about the source of the file.

Table 144: 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 145: 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] section 7.1
2. Variable Header: It contains the information for of the reference frame of the state vectors in the file. The format is in table 146.
3. Data Block: It consists in a set of structures described in the tables below:

Table 146: 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 QUASI_TRUE_DATE

Table 147: Predicted Orbit File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_OSVs	List of <OSV> Structures (See table 148)	count="n" where n is the number of elements in the list	-	List of Orbit State Vectors

Table 148: Predicted 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

Table 148: Predicted Orbit File. OSV

Tag name	type	Attribute	C Format	Description
Absolute_Orbit	integer		%+06ld	absolute orbit counter This counter is incremented by one unit from a record to the next. It must be differentiated 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 system
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 "000000000000"

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_ORBPRES_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>CS_OPER_MPL_ORBPRES_20020315T205400_20020321T205500_0001</File_Name>
      <File_Description>FOS Predicted Orbit File</File_Description>
      <Notes></Notes>
      <Mission>CryoSat</Mission>
      <File_Class>Routine Operations</File_Class>
      <File_Type>MPL_ORBPRES</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>
  </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] section 7.1
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 149: Orbit Scenario File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_Orbit_Changes	List of <Orbit_Change> Structures (See table 150)	count="n" where n is the number of elements in the list	-	List of Orbital Changes

Table 150: Orbit Scenario File. Orbit_Change

Tag name	type	Attribute	C Format	Description
Orbit	structure (see table 151)			Orbit information
Cycle	structure (see table 152)			Cycle information
Time_of_ANX	structure (see table 153)			Ascending node time

Table 151: 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 152: 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 (ANX)
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

Table 153: Orbit Scenario File. Time_of_ANX

Tag name	type	Attribute	C Format	Description
TAI	date		%s	TAI date and time of ANX, in ASCII CCSDS time format, including time reference and micro-seconds
UTC	date		%s	UTC date and time of ANX, in ASCII CCSDS time format, including time reference and micro-seconds
UT1	date		%s	UT1 date and time of ANX, in ASCII CCSDS time format, including time reference 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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>CS_OPER_MPL_ORBSCT_20020312T140002_99999999T999999_0001</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>
      </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>
      </Orbit_Change>
    </List_of_Orbit_Changes>
  </Data_Block>
</Earth_Explorer_File>
```



```

        <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>
    </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>
    </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

9.6.1 Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT] section 7.1
2. Variable Header: It contains the information for of the reference frame of the state vectors in the file. The format is in table 146.
3. Data Block: It consists in a set of structures described in the tables below:

Table 154: Orbit Scenario File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_Orbit_Changes	List of <Orbit_Change> Structures (See table 150)	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 148)	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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>CS_TEST_MPL_ORBREF_20020303T080916_20020303T094823_0001</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>EXPLORER_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>
    </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>
    </List_of_Orbit_Changes>
  </Data_Block>
</Earth_Explorer_File>
```

```

</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] section 7.1
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 155: Satellite Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
Satellite_Name	string	-	%s	Satellite Name
NORAD_Data	structure (see table 157)	-	-	NORAD Satellite data
Lib_Init	structure (see table 157)	-	-	Low and tight tolerances for orbital parameters
Orbit_Init	structure (see table 158)	-	-	Default Orbital parameters

Table 156: 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 157: Satellite Configuration File. Lib_Init Structure

Tag name	type	Attribute	C Format	Description
Low_Tolerances	structure (see table 159)	-	-	Low tolerances for orbital parameters
Tight_Tolerances	structure (see table 155)	-	-	Tight tolerances for orbital parameters

Table 158: 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)

Table 158: Satellite Configuration File. Orbit_InitStructure

Tag name	type	Attribute	C Format	Description
Max_Inclination	real	-	%lf	Maximum inclination (degrees)
Nom_Eccentricity	real	-	%lf	Nominal Eccentricity
Nom_Arg_Perigee	real	-	%lf	Nominal Argument of perigee (degrees)

Table 159: 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="01.00">
  <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] section 7.1
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 160: Attitude File. Data Block

Tag name	type	Attribute	C Format	Description
Attitude_File_Type	string	-	%s	The initial attitude frame. It can be: <ul style="list-style-type: none"> • Sat_Nominal_Attitude • Sat_Attitude • Instr_Attitude
Attitude_Data_Type	string	-	%s	It defines the type of attitude data: <ul style="list-style-type: none"> • 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 161 for the angles data or table 162 for the quaternions	-	-	Structure for the list of angles or the quaternions

Table 161: Attitude File. Attitude Angles Data

Tag name	type	Attribute	C Format	Description
List_of_Attitude_Angles	List of Attitude_Angles (see table 163)	count="n"	-	List of Attitude_Angles

Table 162: 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 QUASI_TRUE_DATE
List_of_Quaternions	List of Quaternions (see table 164)	count="n"	-	List of Quaternions

Table 163: Attitude File. List of Attitude Angles

Tag name	type	Attribute	C Format	Description
Attitude_Angles	Structure (see table 165)	-	-	Pitch, roll and yaw angles for a given time

Table 164: Attitude File. List of Quaternions Data

Tag name	type	Attribute	C Format	Description
Quaternions	Structure (see table 166)	-	-	Set of quaternions for a given time

Table 165: Attitude File. Attitude_Angles

Tag name	type	Attribute	C Format	Description
Time	string	ref="RRR" where RRR stands for: <ul style="list-style-type: none"> • 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 166: Attitude File. Quaternions

Tag name	type	Attribute	C Format	Description
Time	string	ref="RRR" where RRR stands for: <ul style="list-style-type: none"> • 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

```
<?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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>ATT_TEST_FILE</File_Name>
      <File_Description>Attitude File</File_Description>
      <Notes/>
      <Mission>XXXXXX</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>
      </List_of_Attitude_Angles>
    </Attitude_Angles_Data>
  </Data_Block>
</Earth_Explorer_File>
```

```

</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>

```

9.11 Star tracker configuration File

9.11.1 Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT] section 7.1
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 167: Star Tracker Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
Satellite_Name	string	-	%s	Satellite Name
Mispointing	Structure (See table 168)	-	-	Set of rotation angles needed for mispointing computation

Table 168: Star Tracker Configuration File. Mispointing

Tag name	type	Attribute	C Format	Description
Aberration_Correction	string	-	%s	Aberration correction flag. Possible values are: <ul style="list-style-type: none"> • Yes: for applying the aberration-correction. • No: for not applying the aberration correction. • Reverse: for applying the aberration correction with the transposed matrix.
Star_Trackers_Limits	Structure (See table 169)	-	-	Limits for the validity fo the quaternions
Star_Trackers_Priority	Structure (See table 170)	-	-	Star trackers priority
List_of_Star_Trackers	Structure (See table 171)	count="n"	-	List of rotation angles from the antenna bench to the star trackers frame
Satellite_Mechanical_To_Antenna_Bench	Structure (See table 172)	-	-	Rotation angles from the satellite mechanical to the antenna bench frame
Satellite_Control_To_Satellite_Mechanical	Structure (See table 173)	-	-	Rotation angles from the satellite control to the satellite mechanical frame
Satellite_Attitude_To_Satellite_Control	Structure (See table 168)	-	-	Rotation angles from the satellite control to the satellite attitude frame

Table 169: Star Tracker Configuration File. Star tracker limits

Tag name	type	Attribute	C Format	Description
Max_Penalty	integer	-	%d	Maximum penalty for the quaternions

Table 169: Star Tracker Configuration File. Star tracker limits

Tag name	type	Attribute	C Format	Description
Quaternion_Norm_Threshold	real	-	%f	Threshold for the modulus of the quaternion
Max_Time_Gap	real	unit="s"	%f	Maximum time gap between two consecutive quaternions

Table 170: 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 171: Star Tracker Configuration File. List_of_Star_Trackers

Tag name	type	Attribute	C Format	Description
Star_Tracker	Structure (See table 172)	-	-	Antenna bench to Star tracker rotation angles

Table 172: Star Tracker Configuration File. Pre and Post Launch angles

Tag name	type	Attribute	C Format	Description
Pre_Launch_Angles	Structure (See table 173)	-	-	pre-launch angles
Post_Launch_Misalignment	Structure (See table 173)	-	-	post-launch angles

Table 173: 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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
  </Earth_Explorer_Header>
  <Data_Block type="xml">
    <Satellite_Name>CryoSat</Satellite_Name>
    <Mispointing>
      <Aberration_Correction>Yes</Aberration_Correction>
    </Mispointing>
  </Data_Block>
</Earth_Explorer_File>
```

```

<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] section 7.1
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 174: DEM Configuration File. Data Block

Tag name	type	Attribute	C Format	Description
The following options are valid: • ACE_Model	Structure (see table 175)	-	-	Structure containing the DEM model. Different choices are possible depending on the DEM model.

Table 175: DEM Configuration File. ACE Model

Tag name	type	Attribute	C Format	Description
Version	long	-	%d	Version of DEM Getasse
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 filenames for DEM files should follow the following convention: xx{N/S}yyy{E/W}.GETASSE30 (where xx is the latitude and yyy the longitude of the southern western point in the file and N, S E and W stands for north, south, east and west respectively)
Interval_X	real	unit="xxx" Possible values: • deg • min • secs	%f	Angular interval between two consecutive points along the X-Axis in a DEM file.
Interval_Y	real	unit="xxx" Possible values: • deg • min • secs	%f	Angular interval between two consecutive points along the Y-Axis in a DEM file.
Num_Points_X	integer	-	%d	Number of points along the X-Axis in a DEM file
Num_Points_Y	integer	-	%d	Number of points along the Y-Axis in a DEM file

Table 175: DEM Configuration File. ACE Model

Tag name	type	Attribute	C Format	Description
Data_Type	string	-	%s	Data type for the altitude values in the dem. Possible values are: <ul style="list-style-type: none"> • int16: for signed integer values of 2 bytes • int32: for signed integer values of 4 bytes • int64: for signed integer values of 8 bytes • uint16: for unsigned integer values of 2 bytes • uint32: for unsigned integer values of 4 bytes • uint64: for unsigned integer values of 8 bytes • float: for real values of 4 bytes • double: for real values of 8 bytes

9.12.2 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_DEMCFG_0102.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>DEM_CONFIG_TEST_FILE</File_Name>
      <File_Description>DEM Configuration File</File_Description>
      <Notes/>
      <Mission></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">
    <ACE_Model>
      <Directory>../../data/DEM_v2</Directory>
      <Interval_X unit="secs">30</Interval_X>
      <Interval_Y unit="secs">30</Interval_Y>
      <Num_Points_X>1800</Num_Points_X>
      <Num_Points_Y>1800</Num_Points_Y>
      <Data_Type>int16</Data_Type>
    </ACE_Model>
  </Data_Block>
</Earth_Explorer_File>
```



Code: EE-MA-DMS-GS-0007
Date: 07/05/2010
Issue: 3.7.3
Page: 178

9.13 Swath Definition File

9.13.1 Format

1. Fixed Header: For the fixed header format, refer to [EE_FMT] section 7.1
2. Variable Header: Empty.
3. Data Block: It consists in a set of structures described in the tables below.

Table 176: Swath Definition File. Data Block

Tag name	type	Attribute	C Format	Description
Swath	Structure (see table 177)	-	-	Swath structure

Table 177: Swath Definition File. Swath

Tag name	type	Attribute	C Format	Description
Output_File_Description	string	-	%s	File Description 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: <ul style="list-style-type: none"> • point • line • inertial
Num_Swath_Records	integer	-	%d	Number of points in the swath template file (>0)
Refraction	Structure (See table 178)	-	-	Refraction model structure
Geometry Union:Choice between the following structures:	List_of_Swath_Points	List of swath geometry (see table 179)	count=n (number of 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 183)	-	ASAR geometry
Sat_Nominal_Att	Structure (see table 184)	-	-	Satellite Nominal Attitude initialization data
Sat_Att	Structure (see table 185)	-	-	Satellite Attitude initialization data
Instr_Att	Structure (see table 185)	-	-	Instrument Attitude initialization data

Table 178: Swath Definition File. Refraction

Tag name	type	Attribute	C Format	Description
Model	string	-	%s	Atmospheric refraction model. It can be one of: <ul style="list-style-type: none"> • NO_REF • STD_REF • USER_REF • PRED_REF
Freq	real	unit="MHz"	%f	Signal Frequency (≥ 0)

Table 179: Swath Definition File. Swath Point

Tag name	type	Attribute	C Format	Description
Swath Point	Union	-	-	Swath point union. It includes one of the tags described in the row below.
Swath_Point Union: Choice between the following structures	Pointing_Geometry	Structure (defined in table 180)	-	Swath geometry according to
	Distance_Geometry	Structure (defined in table 182)	-	
	Limb_Geometry	Structure (defined in table 181)	-	
	Inertial_Geometry	Structure (defined in table 181)	-	
	Sub Satellite Geometry	empty tag	-	

Table 180: 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 181: 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 182: 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 183: Swath Definition File. ASAR Geometry

Tag name	type	Attribute	C Format	Description
Left_Pt	Structure (defined in table 180)	-	-	Left Point
Mid_Pt	Structure (defined in table 180)	-	-	Middle Point
Right_Pt	Structure (defined in table 180)	-	-	Right Point
Left_Slant_Range_Ext	real	unit="10-6s"	%f	Slant Range Extension parameter for left point. In case of narrow ASAR swath only this parameter is used.
Right_Slant_Range_Ext	real (optional parameter)	unit="10-6s"	%f	Slant Range Extension parameter for right point. Only used for wide ASAR swaths.

Table 184: Swath Definition File. Satellite Nominal Attitude

Tag name	type	Attribute	C Format	Description	
Choice between one of the following tags:	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 186)	-	-	Attitude initialization with parameter model
	Harmonic_Model	Structure (see table 187)	-	-	Attitude initialization with harmonic model
	File_Model	Structure (see table 188)	-	-	Attitude initialization with a data file

Table 185: 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 187)	-	-	Attitude initialization with harmonic model
	File_Model	Structure (see table 188)	-	-	Attitude initialization with a data file
	Angle_Model	Structure (see table 189)	-	-	Attitude initialization with angles
	Matrix_Model	Structure (see table 190)	-	-	Attitude initialization with a Matrix

Table 186: Swath Definition File. Parameter Model

Tag name	type	Attribute	C Format	Description
Model	integer	-	%d	Parameter model
List_of_Parameters	List of <i><Parameter></i> structures (see table 191)	count="n" where n is the number of elements in the list	-	List of parameters as used for the CFI function <i>xp_sat_nominal_att_init_model</i> (See [POINT_SUM])

Table 187: Swath Definition File. Harmonic Model

Tag name	type	Attribute	C Format	Description
Angle_Type	integer	-	%d	Angle type
List_of_Harmonic_s_Pitch	List of <i><Harmonic_Pitch></i> structures (see table 192)	count="n" where n is the number of elements in the list	-	List of harmonic pitch coefficients
List_of_Harmonic_Roll	List of <i><Harmonic_Roll></i> structures (see table 192)	count="n" where n is the number of elements in the list	-	List of harmonic roll coefficients
List_of_Harmonic_Yaw	List of <i><Harmonic_Yaw></i> structures (see table 192)	count="n" where n is the number of elements in the list	-	List of harmonic yaw coefficients
Offsets (only for Instr_Att element)	Structure (see table 194)	-	-	Offsets

Table 188: Swath Definition File. File Model

Tag name	type	Attribute	C Format	Description
List_of_Files	List of <File> (see table 195)	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 196)	-	-	It indicates the time window to be read from the attitude files.

Table 189: 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 194)	-	-	Offsets

Table 190: Swath Definition File. Matrix Model

Tag name	type	Attribute	C Format	Description
Row_1	Structure (see table 198)	-	-	Mispointing matrix first row
Row_2	Structure (see table 198)	-	-	Mispointing matrix second row
Row_3	Structure (see table 198)	-	-	Mispointing matrix third row
Offsets (only for Instr_Att element)	Structure (see table 194)	-	-	Offsets

Table 191: Swath Definition File. List_of_Parameters

Tag name	type	Attribute	C Format	Description
Parameter	string	-	%s	Parameter

Table 192: Swath Definition File. List_of_Harmonics_Pitch/Roll/Yaw

Tag name	type	Attribute	C Format	Description
Harmonic	Structure (see table 193)	-	-	Harmonic structure

Table 193: Swath Definition File. Harmonic

Tag name	type	Attribute	C Format	Description
Harmonic_Type	integer	-	%d	Harmonic type
Harmonic_Coef	real	-	%f	Harmonic coefficient

Table 194: 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 195: Swath Definition File. File

Tag name	type	Attribute	C Format	Description
File	string	-	-	Attitude filename (complete path)

Table 196: 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 197)	-	-	A time window will be read from the files

Table 197: 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 198: 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

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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <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] section 7.1
2. Variable Header: It consists in a set of structures described in the tables below.

Table 199: Swath Template File. Variable_Header

Tag name		type	Attribute	C Format	Description
Swath_Def_File		string	-	%s	Swath definition file used for generating the file
Swath_Type		string. It can have one of the following values: <ul style="list-style-type: none"> • open • closed 	-	%s	Swath type
Swath_Point_Type		string. It can have one of the following values: <ul style="list-style-type: none"> • 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 201)	-	-	Set of orbital parameters
	Orbit_State_Vector	Structure (see table 202)	-	-	Orbit state vector
Time_Step		real	unit="s"	%f	Seconds between two swath points
List_of_STF_Altitudes		List of <STF Altitude> (see table 200)	count="n" where n is the number of elements in the list (n>=1)	-	
Refraction		Structure (see table 204)	-	-	Refraction data

Table 200: Swath Template File. STF_Altitude

Tag name	type	Attribute	C Format	Description
STF_Altitude	real	unit="m"	%f	Altitude for a swath point

Table 201: Swath Template File. Orbit_Geometry

Tag name	type	Attribute	C Format	Description
Repeat_Cycle	real	unit="day"	%f	Repeat cycle in days

Table 201: Swath Template File. Orbit_Geometry

Tag name	type	Attribute	C Format	Description
Cycle_Length	real	unit="orbit"	%f	Cycle length in orbits
MLST_Drift	real	unit="s/day"	%f	Mean local solar time drift

Table 202: 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 Y coordinate (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 203: 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 204: Swath Template File. Refraction

Tag name	type	Attribute	C Format	Description
Model	string	-	%s	Atmospheric refraction model. It can be one of: <ul style="list-style-type: none"> • 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 205: Swath Template File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_STF_Pts	List of <STF_Pt> (See table 206)	count="n" where n is the number of elements in the list	-	List of points in the swath

Table 206: 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 207)	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 208)		

Table 207: 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 208: 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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>LINE_SWATH_FILE.XML</File_Name>
      <File_Description>Swath Template File</File_Description>
      <Notes/>
      <Mission>XXXXXX</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>
<Swath_Def_File></Swath_Def_File>
<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_Alitudes count="4">
  <STF_Alitude unit="m">+000000.000</STF_Alitude>
  <STF_Alitude unit="m">+000000.000</STF_Alitude>
  <STF_Alitude unit="m">+000000.000</STF_Alitude>
  <STF_Alitude unit="m">+000000.000</STF_Alitude>
</List_of_STF_Alitudes>
<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>
  </List_of_STF_Pts count="1200">

```

[...]

```
<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] section 7.1
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 209: Zone Database File. Data_Block

Tag name	type	Attribute	C Format	Description
List_of_Zones	List of <Zone> Structures (See table 210)	count="n" where n is the number of elements in the list	-	List of zones

Table 210: 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	Creator name
List_of_Polygon_Pts	list of structures <Polygon_Pt> (See table 211)	count="n" where n is the number of elements 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 empty.

Table 211: 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_0100.XSD"
  xmlns="http://eop-cfi.esa.int/CFI"
  schemaVersion="01.00">
  <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>
    </Zone>
  </List_of_Zones>
</Data_Block>

```

```

    </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] section 7.1
2. Variable Header: Empty
3. Data Block: It consists in a set of structures described in the tables below:

Table 212: 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 213)	-	-	Station location
List_of_Spacecrafts	List of <Spacecraft> structures (see table 215)	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 214)	count="n" where n is the number of elements in the list	-	Mask points

Table 213: 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 214: Station Database File. Mask_Point

Tag name	type	Attribute	C Format	Description
Az	real	unit="deg"	%f	Azimuth
El	real	unit="deg"	%f	Elevation

Table 215: 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

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_0100.XSD"
xmlns="http://eop-cfi.esa.int/CFI"
schemaVersion="01.00">
  <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>
      </Ground_Station>
    </List_of_Ground_Stations count="n">
    <List_of_Spacecrafts count="1">
      <Spacecraft>
        <Name>SMOS</Name>
        <Aos_El unit="deg">+10.0</Aos_El>
      </Spacecraft>
    </List_of_Spacecrafts count="1">
  </Data_Block>
</Earth_Explorer_File>
```

```

    <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>

10 LIBRARY PRECAUTIONS

The following precaution shall be taking into account when using EXPLORER_DATA_HANDLING library:

- None

11 KNOWN PROBLEMS

The following precautions shall be taken into account when using the CFI software libraries:

Table 216: Known problems

CFI library	Problem	Work around solution
-	-	-