

**Earth Observation  
Mission CFI Software  
EO\_LIB  
SOFTWARE USER MANUAL**

**Code:** EO-MA-DMS-GS-0003  
**Issue:** 4.3  
**Date:** 06/02/12

|                     | <b>Name</b>                   | <b>Function</b>  | <b>Signature</b> |
|---------------------|-------------------------------|------------------|------------------|
| <b>Prepared by:</b> | Mariano Sánchez-Nogales       | Project Engineer |                  |
|                     | Fabrizio Pirondini            | Project Engineer |                  |
|                     | Juan Jose Borrego Bote        | Project Engineer |                  |
|                     | Carlos Villanueva             | Project Engineer |                  |
|                     | Rubén Castro                  | Project Engineer |                  |
| <b>Checked by:</b>  | José Antonio González Abeytua | Project Manager  |                  |
| <b>Approved by:</b> | 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

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       |   |
|                  |                | Public         |   |
| Contract Issuer: | ESA / ESTEC    | Industry       | X |
|                  |                | Confidential   |   |

| External Distribution |              |        |
|-----------------------|--------------|--------|
| Name                  | Organisation | Copies |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |
|                       |              |        |

| Electronic handling   |                      |
|-----------------------|----------------------|
| Word Processor:       | OpenOffice 2.0       |
| Archive Code:         | P/SUM/DMS/01/026-008 |
| Electronic file name: | eo-ma-dms-gs-003-21  |

## DOCUMENT STATUS LOG

| Issue | Change Description  | Date     | Approval |
|-------|---|----------|----------|
| 1.0   | New document  | 08/11/01 |          |
| 1.1   | Updated Time Transformation functions   | 04/02/02 |          |
| 1.2   | Updated the following functions:<br>xl_change_cart_cs, xl_geod_to_cart,<br>xl_cart_to_geod, xl_kepl_to_cart, xl_cart_to_kepl,<br>xl_sun, xl_moon, xl_planet, xl_star_radec,<br>xl_geod_distance, xl_time_ref_init_file,<br>xl_time_ref_close.<br>The xl_attitude_cs function has been removed<br>and replaced by xp_attitude in the<br>EXPLORER_POINTING library. | 15/04/02 |          |
| 1.3   | Added xl_time_get_leap_second_info  | 19/07/02 |          |
| 2.0   | Maintenance release.  | 29/11/02 |          |
| 2.1   | Maintenance release.  | 13/05/03 |          |
| 2.2   | Added xl_default_sat_init function.   | 30/09/03 |          |
| 3.0   | New initialisation strategy and interfaces  | 21/07/04 |          |
| 3.1   | Maintenance Release.<br>New functions:<br>- xl_get_rotation_angles,<br>- xl_get_rotated_vectors,<br>- xl_position_on_orbit  | 13/10/04 |          |
| 3.2   | Maintenance release.  | 15/11/04 |          |
| 3.3   | Maintenance release.<br>New features:<br>- Changes for dealing with the new library<br>explorer_data_handling<br>- Identifier accessors.<br>- OBT to UTC conversion for ADM and SMOS<br>- Support for ENVISAT ASCII files removed   | 11/07/05 |          |
| 3.4   | Maintenance release.<br>New function xl_default_sat_close.  | 18/11/05 |          |

|     |  |          |  |
|-----|--|----------|--|
| 3.5 | <p>Maintenance release.<br/>         New features for xl_time_ref_init_file.<br/>         New features for xl_change_cart_cs<br/>         New functions prototypes:<br/>         - xl_cart_to_radec<br/>         - xl_radec_to_cart<br/>         - xl_star_catalog<br/>         - xl_topocentric_to_ef<br/>         - xl_ef_to_topocentric</p>   | 26/05/06 |  |
| 3.6 | <p>Maintenance release.<br/>         New features:<br/>         - xl_change_cart_cs<br/>         - SMOS UTC proteus time format<br/>         - Parameters for SENTINEL-1<br/>         New functions implemented:<br/>         - xl_euler_to_matrix and xl_matrix_to_euler<br/>         - xl_cart_to_radec and xl_radec_to_cart<br/>         - xl_star_catalog<br/>         - xl_topocentric_to_ef and xl_ef_to_topocentric</p> | 24/11/06 |  |
| 3.7 | <p>Maintenance release.<br/>         New features:<br/>         - Function expcfi_check_libs<br/>         - Library version for MAC OS X on Intel (32 and 64-bits)</p>   | 13/07/07 |  |
| 3.8 | <p>Maintenance release.<br/>         New features:<br/>         - Parameters for SENTINEL-2, SENTINEL-3 and SEOSAT<br/>         - Generic Satellite</p>  | 31/07/08 |  |
| 4.0 | <p>Maintenance release.<br/>         New features:<br/>         - Function interfaces changed for model support</p>  | 19/01/09 |  |
| 4,1 | <p>Maintenance release.<br/>         New features:<br/>         - Time initialization with list of files<br/>         - Time initialization with OSF</p>   | 07/05/10 |  |
| 4.2 | <p>Maintenance release.</p>  | 31/01/11 |  |

|     |  |  |  |
|-----|--|--|--|
| 4.3 | Maintenance release.<br>New features: <ul style="list-style-type: none"><li>- Pseudo-EF CS added.</li><li>- Polar motion included in EF CS.</li><li>- New init function xl_time_id_init.</li><li>- TimeId initialization support IERS Bulletin A and Bulletins A+B</li><li>- New time transport formats:<ul style="list-style-type: none"><li>- XL_TRANS_GENERIC_GPS</li><li>- XL_TRANS_GENERIC_GPS_WEEK</li></ul></li></ul> |  |  |
|-----|--|--|--|

## TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>DOCUMENT INFORMATION</b> .....                          | <b>2</b>  |
| <b>DOCUMENT STATUS LOG</b> .....                           | <b>3</b>  |
| <b>TABLE OF CONTENTS</b> .....                             | <b>6</b>  |
| <b>LIST OF TABLES</b> .....                                | <b>18</b> |
| <b>LIST OF FIGURES</b> .....                               | <b>24</b> |
| <b>1 SCOPE</b> .....                                       | <b>25</b> |
| <b>2 ACRONYMS, NOMENCLATURE AND TERMINOLOGY</b> .....      | <b>26</b> |
| 2.1 Acronyms.....  | 26        |
| 2.2 Nomenclature.....                                      | 26        |
| 2.3 Note on Terminology.....                               | 27        |
| <b>3 APPLICABLE AND REFERENCE DOCUMENTS</b> .....          | <b>28</b> |
| 3.1 Applicable Documents.....                              | 28        |
| 3.2 Reference Documents.....                               | 28        |
| <b>4 INTRODUCTION</b> .....                                | <b>29</b> |
| 4.1 Functions Overview.....                                | 29        |
| 4.1.1 Time Computations.....                               | 29        |
| 4.1.1.1 Time Reference Transformations Initialization..... | 29        |
| 4.1.1.2 Time Format and Reference Transformations.....     | 29        |
| 4.1.1.3 Operation between Dates.....                       | 30        |
| 4.1.1.4 Transformations from/to On-board Times.....        | 30        |
| 4.1.2 Coordinate Systems Transformations.....              | 30        |
| 4.1.2.1 Reference Frames Transformations.....              | 30        |
| 4.1.2.2 Attitude-related Computations.....                 | 30        |

---

|            |   |           |
|------------|---|-----------|
| 4.1.2.3    | Coordinates Transformations.....                                    | 31        |
| 4.1.2.4    | State Vector Transformations.....                                   | 31        |
| 4.1.2.5    | Position on orbit calculations.....                                 | 31        |
| 4.1.3      | Other Basic Computations.....                                       | 31        |
| 4.1.4      | Astronomical model selection.....                                   | 31        |
| <b>4.2</b> | <b>Time Reference Transformations Calling Sequence.....</b>         | <b>31</b> |
| <b>4.3</b> | <b>Earth and Astronomical model selection calling sequence.....</b> | <b>33</b> |
| <b>5</b>   | <b>LIBRARY INSTALLATION.....</b>                                    | <b>35</b> |
| <b>6</b>   | <b>LIBRARY USAGE.....</b>   | <b>36</b> |
| <b>6.1</b> | <b>Usage hints.....</b>   | <b>38</b> |
| <b>6.2</b> | <b>General Enumerations.....</b>                                    | <b>39</b> |
| <b>6.3</b> | <b>Data Structures.....</b>   | <b>47</b> |
| <b>7</b>   | <b>CFI FUNCTIONS DESCRIPTION.....</b>                               | <b>52</b> |
| <b>7.1</b> | <b>xl_time_ref_init_file.....</b>                                   | <b>53</b> |
| 7.1.1      | Overview.....   | 53        |
| 7.1.2      | Calling interface.....  | 54        |
| 7.1.3      | Input parameters.....   | 55        |
| 7.1.4      | Output parameters.....  | 57        |
| 7.1.5      | Warnings and errors.....  | 58        |
|            | .....   | 59        |
| <b>7.2</b> | <b>xl_time_ref_init.....</b>  | <b>60</b> |
| 7.2.1      | Overview.....   | 60        |
| 7.2.2      | Calling interface.....  | 60        |
| 7.2.3      | Input parameters.....   | 60        |
| 7.2.4      | Output parameters.....  | 61        |
| 7.2.5      | Warnings and errors.....  | 61        |
| <b>7.3</b> | <b>xl_time_id_init .....</b>  | <b>63</b> |
| 7.3.1      | Overview .....  | 63        |
| 7.3.2      | Calling interface .....   | 63        |

---

---

|                                     |           |
|-------------------------------------|-----------|
| 7.3.3 Input parameters.....         | 63        |
| 7.3.4 Output parameters.....        | 64        |
| 7.3.5 Warnings and errors .....     | 65        |
| <b>7.4 xl_time_close.....</b>       | <b>67</b> |
| 7.4.1 Overview.....                 | 67        |
| 7.4.2 Calling interface.....        | 67        |
| 7.4.3 Input parameters.....         | 67        |
| 7.4.4 Output parameters.....        | 67        |
| 7.4.5 Warnings and errors.....      | 68        |
| <b>7.5 xl_time_get_id_data.....</b> | <b>69</b> |
| 7.5.1 Overview.....                 | 69        |
| 7.5.2 Calling interface.....        | 69        |
| 7.5.3 Input parameters.....         | 69        |
| 7.5.4 Output parameters.....        | 69        |
| 7.5.5 Warnings and errors.....      | 70        |
| <b>7.6 xl_time_set_id_data.....</b> | <b>71</b> |
| 7.6.1 Overview.....                 | 71        |
| 7.6.2 Calling interface.....        | 71        |
| 7.6.3 Input parameters.....         | 71        |
| 7.6.4 Output parameters.....        | 71        |
| 7.6.5 Warnings and errors.....      | 72        |
| <b>7.7 xl_run_init.....</b>         | <b>73</b> |
| 7.7.1 Overview.....                 | 73        |
| 7.7.2 Calling interface.....        | 73        |
| 7.7.3 Input parameters.....         | 73        |
| 7.7.4 Output parameters.....        | 73        |
| 7.7.5 Warnings and errors.....      | 74        |
| <b>7.8 xl_run_get_ids.....</b>      | <b>75</b> |
| 7.8.1 Overview.....                 | 75        |
| 7.8.2 Calling interface.....        | 75        |
| 7.8.3 Input parameters.....         | 75        |
| 7.8.4 Output parameters.....        | 75        |

---



---

|   |           |
|---|-----------|
| 7.8.5 Warnings and errors.....                    | 76        |
| <b>7.9 xl_run_close.....</b>                      | <b>77</b> |
| 7.9.1 Overview.....                               | 77        |
| 7.9.2 Calling interface.....                      | 77        |
| 7.9.3 Input parameters.....                       | 77        |
| 7.9.4 Output parameters.....                      | 77        |
| 7.9.5 Warnings and errors.....                    | 77        |
| <b>7.10 xl_time_ascii_to_ascii.....</b>           | <b>78</b> |
| 7.10.1 Overview.....                              | 78        |
| 7.10.2 Calling Interface.....                     | 78        |
| 7.10.3 Input Parameters.....                      | 79        |
| 7.10.4 Output Parameters.....                     | 79        |
| 7.10.5 Warnings and Errors.....                   | 79        |
| <b>7.11 xl_time_ascii_to_processing.....</b>      | <b>82</b> |
| 7.11.1 Overview.....                              | 82        |
| 7.11.2 Calling Interface.....                     | 82        |
| 7.11.3 Input Parameters.....                      | 83        |
| 7.11.4 Output Parameters.....                     | 83        |
| 7.11.5 Warnings and Errors.....                   | 84        |
| <b>7.12 xl_time_ascii_to_transport.....</b>       | <b>86</b> |
| 7.12.1 Overview.....                              | 86        |
| 7.12.2 Calling Interface.....                     | 86        |
| 7.12.3 Input Parameters.....                      | 87        |
| 7.12.4 Output Parameters.....                     | 87        |
| 7.12.5 Warnings and Errors.....                   | 88        |
| <b>7.13 xl_time_processing_to_ascii.....</b>      | <b>90</b> |
| 7.13.1 Overview.....                              | 90        |
| 7.13.2 Calling Interface.....                     | 90        |
| 7.13.3 Input Parameters.....                      | 90        |
| 7.13.4 Output Parameters.....                     | 91        |
| 7.13.5 Warnings and Errors.....                   | 91        |
| <b>7.14 xl_time_processing_to_processing.....</b> | <b>94</b> |

---

---

|  |            |
|--|------------|
| 7.14.1 Overview.....                             | 94         |
| 7.14.2 Calling Interface.....                    | 94         |
| 7.14.3 Input Parameters.....                     | 95         |
| 7.14.4 Output Parameters.....                    | 95         |
| 7.14.5 Warnings and Errors.....                  | 95         |
| <b>7.15 xl_time_processing_to_transport.....</b> | <b>97</b>  |
| 7.15.1 Overview.....                             | 97         |
| 7.15.2 Calling Interface.....                    | 97         |
| 7.15.3 Input Parameters.....                     | 98         |
| 7.15.4 Output Parameters.....                    | 98         |
| 7.15.5 Warnings and Errors.....                  | 99         |
| <b>7.16 xl_time_transport_to_ascii.....</b>      | <b>100</b> |
| 7.16.1 Overview.....                             | 100        |
| 7.16.2 Calling Interface.....                    | 100        |
| 7.16.3 Input Parameters.....                     | 101        |
| 7.16.4 Output Parameters.....                    | 101        |
| 7.16.5 Warnings and Errors.....                  | 101        |
| <b>7.17 xl_time_transport_to_processing.....</b> | <b>104</b> |
| 7.17.1 Overview.....                             | 104        |
| 7.17.2 Calling Interface.....                    | 104        |
| 7.17.3 Input Parameters.....                     | 105        |
| 7.17.4 Output Parameters.....                    | 105        |
| 7.17.5 Warnings and Errors.....                  | 106        |
| <b>7.18 xl_time_transport_to_transport.....</b>  | <b>108</b> |
| 7.18.1 Overview.....                             | 108        |
| 7.18.2 Calling Interface.....                    | 108        |
| 7.18.3 Input Parameters.....                     | 108        |
| 7.18.4 Output Parameters.....                    | 109        |
| 7.18.5 Warnings and Errors.....                  | 109        |
| <b>7.19 xl_time_add.....</b>                     | <b>111</b> |
| 7.19.1 Overview.....                             | 111        |
| 7.19.2 Calling interface.....                    | 111        |

---

|                                      |            |
|--------------------------------------|------------|
| 7.19.3 Input parameters.....         | 111        |
| 7.19.4 Output parameters.....        | 112        |
| 7.19.5 Warnings and errors.....      | 112        |
| <b>7.20 xl_time_diff.....</b>        | <b>114</b> |
| 7.20.1 Overview.....                 | 114        |
| 7.20.2 Calling interface.....        | 114        |
| 7.20.3 Input parameters.....         | 114        |
| 7.20.4 Output parameters.....        | 115        |
| 7.20.5 Warnings and errors.....      | 115        |
| <b>7.21 xl_time_obt_to_time.....</b> | <b>117</b> |
| 7.21.1 Overview.....                 | 117        |
| 7.21.2 Calling interface.....        | 119        |
| 7.21.3 Input parameters.....         | 120        |
| 7.21.4 Output parameters.....        | 122        |
| 7.21.5 Warnings and errors.....      | 123        |
| <b>7.22 xl_time_time_to_obt.....</b> | <b>125</b> |
| 7.22.1 Overview.....                 | 125        |
| 7.22.2 Calling interface.....        | 127        |
| 7.22.3 Input parameters.....         | 128        |
| 7.22.4 Output parameters.....        | 130        |
| 7.22.5 Warnings and errors.....      | 131        |
| <b>7.23 xl_change_cart_cs.....</b>   | <b>133</b> |
| 7.23.1 Overview.....                 | 133        |
| 7.23.2 Calling interface.....        | 133        |
| 7.23.3 Input parameters.....         | 134        |
| 7.23.4 Output parameters.....        | 135        |
| 7.23.5 Warnings and errors.....      | 135        |
| <b>7.24 xl_geod_to_cart.....</b>     | <b>137</b> |
| 7.24.1 Overview.....                 | 137        |
| 7.24.2 Calling interface.....        | 137        |
| 7.24.3 Input parameters.....         | 137        |
| 7.24.4 Output parameters.....        | 138        |

---

|                                       |            |
|---------------------------------------|------------|
| 7.24.5 Warnings and errors.....       | 138        |
| <b>7.25 xl_cart_to_geod.....</b>      | <b>140</b> |
| 7.25.1 Overview.....                  | 140        |
| 7.25.2 Calling interface.....         | 140        |
| 7.25.3 Input parameters.....          | 140        |
| 7.25.4 Output parameters.....         | 141        |
| 7.25.5 Warnings and errors.....       | 141        |
| <b>7.26 xl_kepl_to_cart.....</b>      | <b>143</b> |
| 7.26.1 Overview.....                  | 143        |
| 7.26.2 Calling interface.....         | 143        |
| 7.26.3 Input parameters.....          | 143        |
| 7.26.4 Output parameters.....         | 144        |
| 7.26.5 Warnings and errors.....       | 144        |
| <b>7.27 xl_cart_to_kepl.....</b>      | <b>146</b> |
| 7.27.1 Overview.....                  | 146        |
| 7.27.2 Calling interface.....         | 146        |
| 7.27.3 Input parameters.....          | 146        |
| 7.27.4 Output parameters.....         | 147        |
| 7.27.5 Warnings and errors.....       | 147        |
| <b>7.28 xl_cart_to_radec.....</b>     | <b>149</b> |
| 7.28.1 Overview.....                  | 149        |
| 7.28.2 Calling interface.....         | 149        |
| 7.28.3 Input parameters.....          | 149        |
| 7.28.4 Output parameters.....         | 150        |
| 7.28.5 Warnings and errors.....       | 150        |
| <b>7.29 xl_radec_to_cart.....</b>     | <b>152</b> |
| 7.29.1 Overview.....                  | 152        |
| 7.29.2 Calling interface.....         | 152        |
| 7.29.3 Input parameters.....          | 152        |
| 7.29.4 Output parameters.....         | 153        |
| 7.29.5 Warnings and errors.....       | 153        |
| <b>7.30 xl_topocentric_to_ef.....</b> | <b>155</b> |

---

---

|                                       |            |
|---------------------------------------|------------|
| 7.30.1 Overview.....                  | 155        |
| 7.30.2 Calling interface.....         | 155        |
| 7.30.3 Input parameters.....          | 155        |
| 7.30.4 Output parameters.....         | 156        |
| 7.30.5 Warnings and errors.....       | 156        |
| <b>7.31 xl_ef_to_topocentric.....</b> | <b>158</b> |
| 7.31.1 Overview.....                  | 158        |
| 7.31.2 Calling interface.....         | 158        |
| 7.31.3 Input parameters.....          | 158        |
| 7.31.4 Output parameters.....         | 159        |
| 7.31.5 Warnings and errors.....       | 159        |
| <b>7.32 xl_sun.....</b>               | <b>161</b> |
| 7.32.1 Overview.....                  | 161        |
| 7.32.2 Calling interface.....         | 161        |
| 7.32.3 Input parameters.....          | 161        |
| 7.32.4 Output parameters.....         | 162        |
| 7.32.5 Warnings and errors.....       | 162        |
| <b>7.33 xl_moon.....</b>              | <b>164</b> |
| 7.33.1 Overview.....                  | 164        |
| 7.33.2 Calling interface.....         | 164        |
| 7.33.3 Input parameters.....          | 164        |
| 7.33.4 Output parameters.....         | 165        |
| 7.33.5 Warnings and errors.....       | 165        |
| <b>7.34 xl_planet.....</b>            | <b>167</b> |
| 7.34.1 Overview.....                  | 167        |
| 7.34.2 Calling interface.....         | 167        |
| 7.34.3 Input parameters.....          | 167        |
| 7.34.4 Output parameters.....         | 168        |
| 7.34.5 Warnings and errors.....       | 168        |
| <b>7.35 xl_star_radec.....</b>        | <b>170</b> |
| 7.35.1 Overview.....                  | 170        |
| 7.35.2 Calling interface.....         | 170        |

---

|   |            |
|---|------------|
| 7.35.3 Input parameters.....                  | 170        |
| 7.35.4 Output parameters.....                 | 171        |
| 7.35.5 Warnings and errors.....               | 172        |
| <b>7.36 xl_star_catalog.....</b>              | <b>173</b> |
| 7.36.1 Overview.....                          | 173        |
| 7.36.2 Calling interface.....                 | 173        |
| 7.36.3 Input parameters.....                  | 173        |
| 7.36.4 Output parameters.....                 | 174        |
| 7.36.5 Warnings and errors.....               | 175        |
| <b>7.37 xl_geod_distance.....</b>             | <b>176</b> |
| 7.37.1 Overview.....                          | 176        |
| 7.37.2 Calling interface.....                 | 176        |
| 7.37.3 Input parameters.....                  | 176        |
| 7.37.4 IOutput parameters.....                | 177        |
| 7.37.5 Warnings and errors.....               | 177        |
| <b>7.38 xl_time_get_leap_second_info.....</b> | <b>179</b> |
| 7.38.1 Overview.....                          | 179        |
| 7.38.2 Calling interface.....                 | 179        |
| 7.38.3 Input parameters.....                  | 180        |
| 7.38.4 Output parameters.....                 | 180        |
| 7.38.5 Warnings and errors.....               | 181        |
| <b>7.39 xl_euler_to_matrix.....</b>           | <b>182</b> |
| 7.39.1 Overview.....                          | 182        |
| 7.39.2 Calling interface.....                 | 182        |
| 7.39.3 Input parameters.....                  | 183        |
| 7.39.4 Output parameters.....                 | 183        |
| 7.39.5 Warnings and errors.....               | 183        |
| <b>7.40 xl_matrix_to_euler.....</b>           | <b>185</b> |
| 7.40.1 Overview.....                          | 185        |
| 7.40.2 Calling interface.....                 | 185        |
| 7.40.3 Input parameters.....                  | 185        |
| 7.40.4 Output parameters.....                 | 186        |

---

---

|  |            |
|--|------------|
| 7.40.5 Warnings and errors.....            | 186        |
| <b>7.41 xl_position_on_orbit.....</b>      | <b>188</b> |
| 7.41.1 Overview.....                       | 188        |
| 7.41.2 Calling interface.....              | 188        |
| 7.41.3 Input parameters.....               | 189        |
| 7.41.4 Output parameters.....              | 189        |
| 7.41.5 Warnings and errors.....            | 190        |
| <b>7.42 xl_get_rotation_angles.....</b>    | <b>191</b> |
| 7.42.1 Overview.....                       | 191        |
| 7.42.2 Calling interface.....              | 191        |
| 7.42.3 Input parameters.....               | 191        |
| 7.42.4 Output parameters.....              | 192        |
| 7.42.5 Warnings and errors.....            | 192        |
| <b>7.43 xl_get_rotated_vectors.....</b>    | <b>194</b> |
| 7.43.1 Overview.....                       | 194        |
| 7.43.2 Calling interface.....              | 194        |
| 7.43.3 Input parameters.....               | 194        |
| 7.43.4 Output parameters.....              | 195        |
| 7.43.5 Warnings and errors.....            | 195        |
| <b>7.44 xl_quaternions_to_vectors.....</b> | <b>197</b> |
| 7.44.1 Overview.....                       | 197        |
| 7.44.2 Calling interface.....              | 197        |
| 7.44.3 Input parameters.....               | 197        |
| 7.44.4 Output parameters.....              | 197        |
| 7.44.5 Warnings and errors.....            | 198        |
| <b>7.45 xl_vectors_to_quaternions.....</b> | <b>199</b> |
| 7.45.1 Overview.....                       | 199        |
| 7.45.2 Calling interface.....              | 199        |
| 7.45.3 Input parameters.....               | 199        |
| 7.45.4 Output parameters.....              | 200        |
| 7.45.5 Warnings and errors.....            | 200        |
| <b>7.46 xl_default_sat_init.....</b>       | <b>201</b> |

---

|                                       |            |
|---------------------------------------|------------|
| 7.46.1 Overview.....                  | 201        |
| 7.46.2 Calling interface.....         | 201        |
| 7.46.3 Input parameters.....          | 201        |
| 7.46.4 Output parameters.....         | 202        |
| 7.46.5 Warnings and errors.....       | 202        |
| <b>7.47 xl_default_sat_close.....</b> | <b>203</b> |
| 7.47.1 Overview.....                  | 203        |
| 7.47.2 Calling interface.....         | 203        |
| 7.47.3 Input parameters.....          | 203        |
| 7.47.4 Output parameters.....         | 203        |
| 7.47.5 Warnings and errors.....       | 203        |
| <b>7.48 xl_set_tle_sat_data.....</b>  | <b>204</b> |
| 7.48.1 Overview.....                  | 204        |
| 7.48.2 Calling interface.....         | 205        |
| 7.48.3 Input parameters.....          | 205        |
| 7.48.4 Output parameters.....         | 205        |
| 7.48.5 Warnings and errors.....       | 206        |
| <b>7.49 xl_model_init.....</b>        | <b>207</b> |
| 7.49.1 Overview.....                  | 207        |
| 7.49.2 Calling interface.....         | 208        |
| 7.49.3 Input parameters.....          | 208        |
| 7.49.4 Output parameters.....         | 209        |
| 7.49.5 Warnings and errors.....       | 209        |
| <b>7.50 xl_model_close.....</b>       | <b>210</b> |
| 7.50.1 Overview.....                  | 210        |
| 7.50.2 Calling interface.....         | 210        |
| 7.50.3 Input parameters.....          | 210        |
| 7.50.4 Output parameters.....         | 210        |
| 7.50.5 Warnings and errors.....       | 210        |
| <b>7.51 xl_model_get_data.....</b>    | <b>212</b> |
| 7.51.1 Overview.....                  | 212        |
| 7.51.2 Calling interface.....         | 212        |



---

|                                       |            |
|---------------------------------------|------------|
| 7.51.3 Input parameters.....          | 212        |
| 7.51.4 Output parameters.....         | 212        |
| 7.51.5 Warnings and errors.....       | 213        |
| <b>8 CFI EXECUTABLE PROGRAMS.....</b> | <b>214</b> |
| 8.1 time_conv.....                    | 214        |
| <b>9 RUNTIME PERFORMANCES.....</b>    | <b>216</b> |
| <b>10 LIBRARY PRECAUTIONS.....</b>    | <b>218</b> |
| <b>11 KNOWN PROBLEMS.....</b>         | <b>219</b> |

## LIST OF TABLES

|  |    |
|--|----|
| Table 1: CFI functions included within EO_LIB library.....             | 37 |
| Table 2: Enumerations within EO_LIB library.....                       | 39 |
| Table 3: Transport time formatsTransport time formatsssssss.....       | 44 |
| Table 4: Basic ASCII time formats.....                                 | 45 |
| Table 5: Derived ASCII time formats.....                               | 45 |
| Table 6: Definition of BOM and EOM for basic ASCII time formats.....   | 46 |
| Table 7: Definition of BOM and EOM for derived ASCII time formats..... | 46 |
| Table 8: EO_LIB structures.....  | 47 |
| Table 9: Time reference correlations from reference files.....         | 53 |
| Table 10: Initialization validity depending on input file.....         | 54 |
| Table 11: Input parameters of xl_time_ref_init_file function.....      | 55 |
| Table 12: Output parameters of xl_time_ref_init_file function.....     | 57 |
| Table 13: Error messages of xl_time_ref_init_file function.....        | 58 |
| Table 14: Input parameters of xl_time_ref_init function.....           | 60 |
| Table 15: Output parameters of xl_time_ref_init function.....          | 61 |
| Table 16: Error messages of xl_time_ref_init function.....             | 61 |
| Table 17: Input parameters of xl_time_id_init function.....            | 63 |
| Table 18: Output parameters of xl_time_id_init function.....           | 64 |
| Table 19: Error messages of xl_time_id_init function.....              | 65 |
| Table 20: Input parameters of xl_time_close function.....              | 67 |
| Table 21: Output parameters of xl_time_close function.....             | 67 |
| Table 22: Error messages of xl_time_close function.....                | 68 |
| Table 23: Input parameters of xl_time_get_id_data function.....        | 69 |
| Table 24: Output parameters of xl_time_get_id_data function.....       | 69 |
| Table 25: Input parameters of xl_time_set_id_data function.....        | 71 |
| Table 26: Output parameters of xl_time_set_id_data function.....       | 71 |
| Table 27: Input parameters of xl_run_init function.....                | 73 |
| Table 28: Output parameters of xl_run_init function.....               | 74 |
| Table 29: Error messages of xl_run_init function.....                  | 74 |
| Table 30: Input parameters of xl_run_get_ids function.....             | 75 |

|  |     |
|--|-----|
| Table 31: Output parameters of xl_run_get_ids function.....                  | 75  |
| Table 32: Input parameters of xl_run_close function.....                     | 77  |
| Table 33: Output parameters of xl_run_close function.....                    | 77  |
| Table 34: Input parameters of xl_time_ascii_to_ascii function.....           | 79  |
| Table 35: Output parameters of xl_time_ascii_to_ascii.....                   | 79  |
| Table 36: Error messages of xl_time_ascii_to_ascii function.....             | 80  |
| Table 37: Input parameters of xl_time_ascii_to_processing function.....      | 83  |
| Table 38: Output parameters of xl_time_ascii_to_processing.....              | 83  |
| Table 39: Error messages of xl_time_ascii_to_processing function.....        | 84  |
| Table 40: Input parameters of xl_time_ascii_to_transport function.....       | 87  |
| Table 41: Output parameters of xl_time_ascii_to_transport.....               | 87  |
| Table 42: Error messages of xl_time_ascii_to_transport function.....         | 88  |
| Table 43: Input parameters of xl_time_processing_to_ascii function.....      | 91  |
| Table 44: Output parameters of xl_time_processing_to_ascii.....              | 91  |
| Table 45: Error messages of xl_time_processing_to_ascii function.....        | 92  |
| Table 46: Input parameters of xl_time_processing_to_processing function..... | 95  |
| Table 47: Output parameters of xl_time_processing_to_processing.....         | 95  |
| Table 48: Error messages of xl_time_processing_to_processing function.....   | 96  |
| Table 49: Input parameters of xl_time_processing_to_transport function.....  | 98  |
| Table 50: Output parameters of xl_time_processing_to_transport.....          | 98  |
| Table 51: Error messages of xl_time_processing_to_transport function.....    | 99  |
| Table 52: Input parameters of xl_time_transport_to_ascii function.....       | 101 |
| Table 53: Output parameters of xl_time_transport_to_ascii.....               | 101 |
| Table 54: Error messages of xl_time_transport_to_ascii function.....         | 102 |
| Table 55: Input parameters of xl_time_transport_to_processing function.....  | 105 |
| Table 56: Output parameters of xl_time_transport_to_processing.....          | 105 |
| Table 57: Error messages of xl_time_transport_to_processing function.....    | 106 |
| Table 58: Input parameters of xl_time_transport_to_transport function.....   | 109 |
| Table 59: Output parameters of xl_time_transport_to_transport.....           | 109 |
| Table 60: Error messages of xl_time_transport_to_transport function.....     | 110 |
| Table 61: Input parameters of xl_time_add function.....                      | 111 |
| Table 62: Output parameters of xl_time_add function.....                     | 112 |
| Table 63: Error messages of xl_time_add function.....                        | 112 |
| Table 64: Input parameters of xl_time_diff function.....                     | 114 |

|   |     |
|---|-----|
| Table 65: Output parameters of xl_time_diff function.....           | 115 |
| Table 66: Error messages of xl_time_diff function.....              | 115 |
| Table 67: Input parameters of xl_time_obt_to_time function.....     | 120 |
| Table 68: Input parameters of xl_envisat_obt_param structure.....   | 120 |
| Table 69: Input parameters of xl_envisat_obt_value structure.....   | 121 |
| Table 70: Input parameters of xl_goce_obt_param structure.....      | 121 |
| Table 71: Input parameters of xl_goce_obt_value structure.....      | 121 |
| Table 72: Input parameters of xl_smos_obt_param structure.....      | 121 |
| Table 73: Input parameters of xl_smos_obt_value structure.....      | 122 |
| Table 74: Input parameters of xl_adm_obt_param structure.....       | 122 |
| Table 75: Input parameters of xl_adm_obt_value structure.....       | 122 |
| Table 76: Output parameters of xl_time_obt_to_time function.....    | 123 |
| Table 77: Error messages of xl_time_obt_to_time function.....       | 123 |
| Table 78: Runtime performances of xl_time_obt_to_time function..... | 123 |
| Table 79: Input parameters of xl_time_obt_to_time function.....     | 128 |
| Table 80: Input parameters of xl_envisat_obt_param structure.....   | 128 |
| Table 81: Input parameters of xl_goce_obt_param structure.....      | 129 |
| Table 82: Input parameters of xl_smos_obt_param structure.....      | 129 |
| Table 83: Input parameters of xl_adm_obt_param structure.....       | 129 |
| Table 84: Output parameters of xl_time_time_to_obt function.....    | 130 |
| Table 85: Output parameters of xl_envisat_obt_value structure.....  | 130 |
| Table 86: Output parameters of xl_goce_obt_value structure.....     | 130 |
| Table 87: Output parameters of xl_smos_obt_value structure.....     | 130 |
| Table 88: Output parameters of xl_adm_obt_value structure.....      | 131 |
| Table 89: Error messages of xl_time_time_to_obt function.....       | 131 |
| Table 90: Runtime performances of xl_time_time_to_obt function..... | 131 |
| Table 91: Input parameters of xl_change_cart_cs function.....       | 134 |
| Table 92: Output parameters of xl_change_cart_cs function.....      | 135 |
| Table 93: Error messages of xl_change_cart_cs function.....         | 136 |
| Table 94: Input parameters of xl_geod_to_cart function.....         | 137 |
| Table 95: Output parameters of xl_geod_to_cart function.....        | 138 |
| Table 96: Error messages of xl_geod_to_cart function.....           | 138 |
| Table 97: Input parameters of xl_cart_to_geod function.....         | 140 |
| Table 98: Output parameters of xl_cart_to_geod function.....        | 141 |

---

|  |     |
|--|-----|
| Table 99: Error messages of xl_cart_to_geod function.....          | 141 |
| Table 100: Input parameters of xl_kepl_to_cart function.....       | 143 |
| Table 101: Output parameters of xl_kepl_to_cart function.....      | 144 |
| Table 102: Error messages of xl_kepl_to_cart function.....         | 144 |
| Table 103: Input parameters of xl_cart_to_kepl function.....       | 146 |
| Table 104: Output parameters of xl_cart_to_kepl function.....      | 147 |
| Table 105: Error messages of xl_cart_to_kepl function.....         | 147 |
| Table 106: Input parameters of xl_cart_to_radec function.....      | 149 |
| Table 107: Output parameters of xl_cart_to_radec function.....     | 150 |
| Table 108: Error messages of xl_cart_to_radec function.....        | 150 |
| Table 109: Input parameters of xl_radec_to_cart function.....      | 152 |
| Table 110: Output parameters of xl_radec_to_cart function.....     | 153 |
| Table 111: Error messages of xl_radec_to_cart function.....        | 153 |
| Table 112: Input parameters of xl_topocentric_to_ef function.....  | 155 |
| Table 113: Output parameters of xl_topocentric_to_ef function..... | 156 |
| Table 114: Error messages of xl_topocentric_to_ef function.....    | 156 |
| Table 115: Input parameters of xl_ef_to_topocentric function.....  | 158 |
| Table 116: Output parameters of xl_ef_to_topocentric function..... | 159 |
| Table 117: Error messages of xl_ef_to_topocentric function.....    | 159 |
| Table 118: Input parameters of xl_sun function.....                | 161 |
| Table 119: Output parameters of xl_sun function.....               | 162 |
| Table 120: Error messages of xl_sun function.....                  | 162 |
| Table 121: Input parameters of xl_moon function.....               | 165 |
| Table 122: Output parameters of xl_moon function.....              | 165 |
| Table 123: Error messages of xl_moon function.....                 | 165 |
| Table 124: Input parameters of xl_planet function.....             | 167 |
| Table 125: Output parameters of xl_planet function.....            | 168 |
| Table 126: Error messages of xl_planet function.....               | 168 |
| Table 127: Input parameters of xl_star_radec function.....         | 171 |
| Table 128: Output parameters of xl_star_radec function.....        | 171 |
| Table 129: Error messages of xl_star_radec function.....           | 172 |
| Table 130: Input parameters of xl_star_catalog function.....       | 173 |
| Table 131: Output parameters of xl_star_catalog function.....      | 174 |
| Table 132: Error messages of xl_star_catalog function.....         | 175 |

|  |     |
|--|-----|
| Table 133: Input parameters of xl_geod_distance function.....              | 176 |
| Table 134: Output parameters of xl_geod_distance function.....             | 177 |
| Table 135: Error messages of xl_geod_distance function.....                | 178 |
| Table 136: Input parameters of xl_time_get_leap_second_info function.....  | 180 |
| Table 137: Output parameters of xl_time_get_leap_second_info function..... | 180 |
| Table 138: Error messages of xl_time_get_leap_second_info function.....    | 181 |
| Table 139: Input parameters of xl_euler_to_matrix function.....            | 183 |
| Table 140: Output parameters of xl_euler_to_matrix function.....           | 183 |
| Table 141: Runtime performances of xl_euler_to_matrix function.....        | 183 |
| Table 142: Input parameters of xl_matrix_to_euler function.....            | 186 |
| Table 143: Output parameters of xl_matrix_to_euler function.....           | 186 |
| Table 144: Error messages of xl_matrix_to_euler function.....              | 186 |
| Table 145: Input parameters of xl_position_on_orbit function.....          | 189 |
| Table 146: Output parameters of xl_position_on_orbit function.....         | 189 |
| Table 147: Error messages of xl_position_on_orbit function.....            | 190 |
| Table 148: Input parameters of xl_get_rotation_angles function.....        | 191 |
| Table 149: Output parameters of xl_get_rotation_angles function.....       | 192 |
| Table 150: Error messages of xl_get_rotation_angles function.....          | 192 |
| Table 151: Input parameters of xl_get_rotated_vectors function.....        | 194 |
| Table 152: Output parameters of xl_get_rotated_vectors function.....       | 195 |
| Table 153: Error messages of xl_get_rotated_vectors function.....          | 195 |
| Table 154: Input parameters of xl_quaternions_to_vectors function.....     | 197 |
| Table 155: Output parameters of xl_quaternions_to_vectors function.....    | 198 |
| Table 156: Error messages of xl_quaternions_to_vectors function.....       | 198 |
| Table 157: Input parameters of xl_vectors_to_quaternions function.....     | 199 |
| Table 158: Output parameters of xl_vectors_to_quaternions function.....    | 200 |
| Table 159: Error messages of xl_vectors_to_quaternions function.....       | 200 |
| Table 160: Input parameters of xl_default_sat_init function.....           | 201 |
| Table 161: Output parameters of xl_default_sat_init function.....          | 202 |
| Table 162: Error messages of xl_default_sat_init function.....             | 202 |
| Table 163: Input parameters of xl_default_sat_close function.....          | 203 |
| Table 164: NORAD Identifiers for satellites.....                           | 204 |
| Table 165: Input parameters of xl_set_tle_sat_data function.....           | 205 |
| Table 166: Possible models for every model type.....                       | 207 |

---

|   |     |
|---|-----|
| Table 167: Model sets.....                                      | 207 |
| Table 168: Input parameters of xl_model_init function.....      | 208 |
| Table 169: Output parameters of xl_model_init function.....     | 209 |
| Table 170: Error messages of xl_model_init function.....        | 209 |
| Table 171: Input parameters of xl_model_close function.....     | 210 |
| Table 172: Output parameters of xl_model_close function.....    | 210 |
| Table 173: Error messages of xl_model_close function.....       | 211 |
| Table 174: Input parameters of xl_model_get_data function.....  | 212 |
| Table 175: Output parameters of xl_model_get_data function..... | 212 |
| Table 176: Known problems.....                                  | 219 |

## LIST OF FIGURES

|  |     |
|--|-----|
| Figure 1: Time reference transformations sequence.....               | 32  |
| Figure 2: Change cartesian coordinates.....                          | 133 |
| Figure 3: Azimuth figures returned by xl_geod_distance function..... | 176 |
| Figure 4: Euler Angles.....  | 182 |



## **1 SCOPE**

The EO\_LIB Software User Manual provides a detailed description of usage of the CFI functions included within the EO\_LIB CFI software library.

## 2 ACRONYMS, NOMENCLATURE AND TERMINOLOGY

### 2.1 Acronyms

|         |  |
|---------|--|
| ANX     | Ascending Node Crossing                            |
| AOCS    | Attitude and Orbit Control Subsystem               |
| ASCII   | American Standard Code for Information Interchange |
| BOM     | Beginning Of Mission                               |
| CFI     | Customer Furnished Item                            |
| EGM96   | Earth Gravitational Model 1996                     |
| EO      | Earth Observation                                  |
| 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) |

## 2.3 Note on Terminology

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

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

## 3 APPLICABLE AND REFERENCE DOCUMENTS

### 3.1 Applicable Documents

[GEN\_SUM] Earth Observation Mission CFI Software. General Software User Manual. EO-MA- DMS-GS-0002. Issue 4.3 06/02/12

### 3.2 Reference Documents

[MCD] Earth Observation Mission CFI Software. Conventions Document. EO-MA-DMS-GS-0001. Issue 4.3 06/02/12

[MSC] Earth Observation Mission CFI Software. Mission Specific Customizations. EO-MA- DMS-GS-0018. Issue 4.3 06/02/12.

[F\_H\_SUM] Earth Observation Mission CFI Software. EO\_FILE\_HANDLING Software User Manual. EO-MA-DMS-GS-0008. Issue 4.3 06/02/12.

[D\_H\_SUM] Earth Observation Mission CFI Software. EO\_DATA\_HANDLING Software User Manual. EO-MA-DMS-GS-007. Issue 4.3 06/02/12.

[IERS] <http://www.iers.org/iers/publications/bulletins/>

## 4 INTRODUCTION

### 4.1 Functions Overview

This software library contains all low-level generic routines, supporting all the other CFI functions.

The following CFI functions are included:

#### 4.1.1 Time Computations

All time time computations are performed internally using the continuous TAI time reference. Therefore the input and output parameters are converted internally to the adequate time reference.

##### 4.1.1.1 Time Reference Transformations Initialization

- **xl\_time\_ref\_init\_file**: initializes time correlations between TAI, UTC, UT1 and GPS times from reference data files.
- **xl\_time\_ref\_init**: initializes time correlations between TAI, UTC, UT1 and GPS times from input reference times.
- **xl\_time\_close**: cleans up any memory allocation performed by the initialization functions.
- **xl\_time\_get\_leap\_second\_info**: retrieves the leap second location (if any) in the initialised time range.

##### 4.1.1.2 Time Format and Reference Transformations

- **xl\_time\_ascii\_to\_ascii**: transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a different ASCII format and/or reference (TAI, UTC, UT1 or GPS).
- **xl\_time\_ascii\_to\_transport**: transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl\_time\_ascii\_to\_processing**: transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl\_time\_processing\_to\_ascii**: transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl\_time\_processing\_to\_transport**: transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl\_time\_processing\_to\_processing**: transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in Processing format with a different reference (TAI, UTC, UT1 or GPS).

- **xl\_time\_transport\_to\_ascii**: transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl\_time\_transport\_to\_transport**: transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in a different Transport format and/or reference (TAI, UTC, UT1 or GPS).
- **xl\_time\_transport\_to\_processing**: transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

#### 4.1.1.3 Operation between Dates

- **xl\_time\_add**: adds a duration to a TAI, UTC, UT1 or GPS time expressed in Processing format.
- **xl\_time\_diff**: subtracts two TAI, UTC, UT1 or GPS times expressed in Processing format.

#### 4.1.1.4 Transformations from/to On-board Times

- **xl\_time\_obt\_to\_time**: transforms an On-board Time (OBT) into a TAI, UTC, UT1 or GPS time in Processing format.
- **xl\_time\_time\_to\_obt**: transforms a TAI, UTC, UT1 or GPS time expressed in Processing format into an On-board Time (OBT).

### 4.1.2 Coordinate Systems Transformations

#### 4.1.2.1 Reference Frames Transformations

- **xl\_change\_cart\_cs**: transforms a state vector between different coordinate systems.
- **xl\_topocentric\_to\_ef**: transforms a state vector from topocentric coordinates to the Earth Fixed CS.
- **xl\_ef\_to\_topocentric**: transforms a state vector from the Earth Fixed CS to topocentric coordinates.

#### 4.1.2.2 Attitude-related Computations

- **xl\_euler\_to\_matrix**: computes the elements of the coordinate transformation matrix with respect to the attitude frame given the corresponding Euler rotation vector in the roll, pitch and yaw sequence.
- **xl\_matrix\_to\_euler**: derives the Euler rotation vector with respect to the attitude frame in the roll, pitch and yaw sequence given the corresponding coordinate transformation matrix.
- **xl\_get\_rotation\_angles**: calculates the rotation angles between two sets of orthonormal right-handed unit vectors expressed wrt an identical coordinate frame.
- **xl\_get\_rotated\_vectors**: calculates the rotated unit vectors given a set of unit vectors and the rotation angles expressed wrt an identical coordinate frame.
- **xl\_quaternions\_to\_vectors**: calculates the orthonormal unit vectors from a given set of quaternions.
- **xl\_vectors\_to\_quaternions**: calculates the set of quaternions that correspond to a set of orthonormal unit vectors.

#### 4.1.2.3 Coordinates Transformations

- **xl\_geod\_to\_cart**: transforms from Geodetic to Cartesian coordinates.
- **xl\_cart\_to\_geod**: transforms from Cartesian to Geodetic coordinates.
- **xl\_cart\_to\_radec**: transforms from a cartesian vector to right ascension and declination.
- **xl\_radec\_to\_cart**: transforms from right ascension and declination to a cartesian vector.

#### 4.1.2.4 State Vector Transformations

- **xl\_kepl\_to\_cart**: transforms from Keplerian to Cartesian coordinates.
- **xl\_cart\_to\_kepl**: transforms from Cartesian to Keplerian coordinates.

#### 4.1.2.5 Position on orbit calculations

- **xl\_position\_on\_orbit**: calculates a value describing the position of the satellite within the orbit, using as input a Cartesian orbit state vector.

### 4.1.3 Other Basic Computations

- **xl\_sun**: calculates the position and velocity of the Sun in the Earth Fixed coordinate system
- **xl\_moon**: calculates the Moon position and velocity in the Earth Fixed coordinate system
- **xl\_planet**: calculates the position and velocity of a selected planet in the Earth Fixed coordinate system
- **xl\_star\_radec**: calculates the right ascension and declination of a star in the True of Date coordinate system.
- **xl\_geod\_distance**: calculates the geodesic distance between two points that lay on the same ellipsoid, and the azimuth of the related geodesic line at both points.
- **xl\_star\_catalog**: calculates the star coordinates in a star catalogue reference frame.

### 4.1.4 Astronomical model selection

- **xl\_model\_init**: It initialises a model identifier that will be used to by other CFI functions to select a model.
- **xl\_model\_close**: cleans up any memory allocation performed by the initialization functions.

## 4.2 Time Reference Transformations Calling Sequence

Time reference transformations ,and other functions with time as input, requires the user to initialise correlations between the different allowed time references, i.e. TAI, UTC, UT1 and GPS time. In order to accomplish such correlations, two possible strategies can be used:

- Initialization from a single or multiple orbit files (**xl\_time\_ref\_init\_file**)
- Initialization with data structures (user data or data read from files) (**xl\_time\_id\_data**).
- Initialization from a given set of time references (**xl\_time\_ref\_init**).

The correlations are stored in a data structure, and the software returns a pointer to it, in addition to the validity range of the initialisation. This structure is referred to as the *Time Id*.

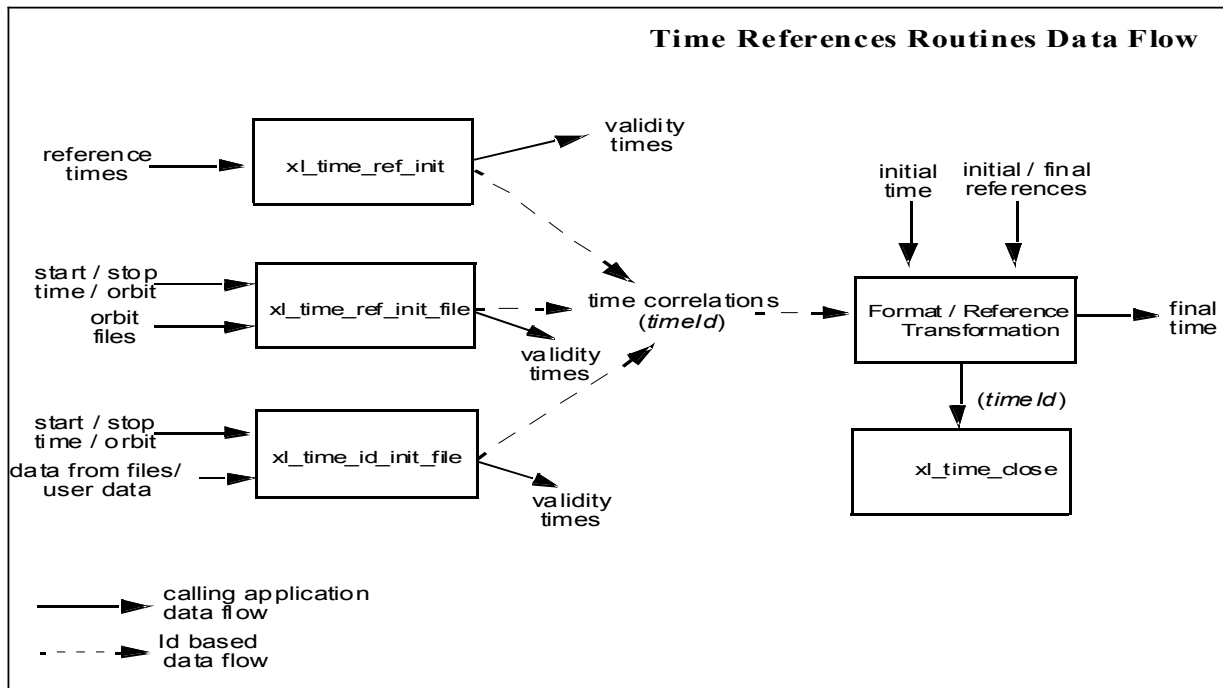
Once the initialisation has been performed, the user is able to transform any date expressed in one of the allowed time references to another, through the Time Format / Reference Transformation functions. The *Time ID* has to be provided to each of these functions. The process can be repeated as needed without initialising the time correlations each time.

After finalising the transformations, the *Time ID* must be freed (**xl\_time\_close**).

A complete view of the time reference transformations sequence is presented in figure 1.

***Figure 1: Time reference transformations sequence***





There is a second way of calling the functions that require a *timeId* as input.

Similar initialisation functions exist in other CFI libraries, resulting in various *Ids* being generated. It is possible to group different *Ids* into a single entity called *runId*. Using this method, a single *runId* can be passed to all functions across the different libraries, instead of passing several *Ids* through the interface.

In this case, the first step would be to create the *timeId*. Then, a *runId* can be generated using as input the *timeId*. This *runId* is then passed through the interface to equivalent functions to those described before (ending in “\_run”).

A detailed description of each function is provided in section 7.

Please refer also to:

- [MCD] for a detailed description of the time references and formats, coordinate systems, parameters and models used in this document.
- [GEN\_SUM] for a complete overview of the CFI, and in particular the detailed description of the Id concept and the error handling functions.

### 4.3 Earth and Astronomical model selection calling sequence

The CFI functions can work with different Earth and astronomical models. These models have been divided in the following categories:

- Star model
- Sun model
- Planet model
- Earth model

- Moon model
- Nutation model
- Precession model
- Constants model

In order to work with different models, these have to be stored in a CFI Id called *Model ID*. The *Model ID* is a variable of type **xl\_model\_id**.

The calling sequence for a C program where the *Model ID* is needed, would be as follows:

- Declare the `model_id` variable:
  - `xl_model_id model_id = {NULL};`
  - The `model_id` has to be initialised this way (as other CFI ID's), so that the EOFCFI could recognise that the `model_id` is not initialised.
- Optionally, initialise the `model_id` with `xl_model_init` function (see section 7.49). This function would set the requested models in the `model_id`. Note that if the `model_id` is not initialised, the EOFCFI functions will use the default models.
- The `model_id` is used as an input parameter in the EOFCFI functions if it is needed.
- Close the `model_id` with `xl_model_close` (Only if the `model_id` was initialised).

Please refer also to:

- [MCD] for a detailed description of the models implemented for the Earth Observation CFI. (For the current version, only the default models are available)
- [GEN\_SUM] for a detailed description of the *Id* concept.

## **5 LIBRARY INSTALLATION**

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

## 6 LIBRARY USAGE

Note that to use the EO\_LIB software library, the following other CFI software libraries are required:

- EO\_FILE\_HANDLING (See [F\_H\_SUM]).
- EO\_DATA\_HANDLING[D\_H\_SUM]

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

- LIBXML2 (MIT license, see [GEN\_SUM]).
- POSIX thread library: libpthread.so (pthread.lib for WINDOWS, with license LGPL)

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

- `explorer_lib.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_lib  
-lexplorer_data_handling -lexplorer_file_handling  
-lxml2 -lpthread
```
- WINDOWS:  

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

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

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

To avoid problems in linking a user application with the EO\_LIB 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 `XL_` or `xl_`.

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

**Table 1: CFI functions included within EO\_LIB library**

| Function Name                    | Enumeration value                   | Long |
|----------------------------------|-------------------------------------|------|
| Main CFI Functions               |                                     |      |
| xl_time_transport_to_ascii       | XL_TIME_TRANSPORT_TO_ASCII_ID       | 0    |
| xl_time_transport_to_transport   | XL_TIME_TRANSPORT_TO_TRANSPORT_ID   | 1    |
| xl_time_transport_to_processing  | XL_TIME_TRANSPORT_TO_PROCESSING_ID  | 2    |
| xl_time_processing_to_ascii      | XL_TIME_PROCESSING_TO_ASCII_ID      | 3    |
| xl_time_processing_to_transport  | XL_TIME_PROCESSING_TO_TRANSPORT_ID  | 4    |
| xl_time_processing_to_processing | XL_TIME_PROCESSING_TO_PROCESSING_ID | 5    |
| xl_time_ascii_to_ascii           | XL_TIME_ASCII_TO_ASCII_ID           | 6    |
| xl_time_ascii_to_transport       | XL_TIME_ASCII_TO_TRANSPORT_ID       | 7    |
| xl_time_ascii_to_processing      | XL_TIME_ASCII_TO_PROCESSING_ID      | 8    |
| xl_time_add                      | XL_TIME_ADD_ID                      | 9    |
| xl_time_diff                     | XL_TIME_DIFF_ID                     | 10   |
| xl_time_obt_to_time              | XL_TIME_OBT_TO_TIME_ID              | 11   |
| xl_time_time_to_obt              | XL_TIME_TIME_TO_OBT_ID              | 12   |
| xl_time_ref_init_file            | XL_TIME_REF_INIT_FILE_ID            | 13   |
| xl_time_ref_init                 | XL_TIME_REF_INIT_ID                 | 14   |
| xl_time_id_init                  | XL_TIME_ID_INIT                     | 15   |
| xl_time_ref_close                | XL_TIME_CLOSE_ID                    | 16   |
| xl_change_cart_cs                | XL_CHANGE_CART_CS_ID                | 17   |
| xl_geod_to_cart                  | XL_GEOD_TO_CART_ID                  | 18   |
| xl_cart_to_geod                  | XL_CART_TO_GEOD_ID                  | 19   |
| xl_kepl_to_cart                  | XL_KEPL_TO_CART_ID                  | 20   |
| xl_cart_to_kepl                  | XL_CART_TO_KEPL_ID                  | 21   |
| xl_sun                           | XL_SUN_ID                           | 22   |
| xl_moon                          | XL_MOON_ID                          | 23   |
| xl_planet                        | XL_PLANET_ID                        | 24   |
| xl_star_radec                    | XL_STAR_RADEC_ID                    | 25   |
| xl_geod_distance                 | XL_GEOD_DISTANCE_ID                 | 26   |
| xl_time_get_leap_second_info     | XL_TIME_GET_LEAP_SECOND_INFO_ID     | 27   |
| xl_default_sat_init              | XL_DEFAULT_SAT_INIT_ID              | 28   |

|                           |                           |    |
|---------------------------|---------------------------|----|
| xl_run_init               | XL_RUN_INIT_ID            | 29 |
| xl_get_rotation_angles    | XL_GET_ROTATION_ANGLES_ID | 30 |
| xl_get_rotated_vectors    | XL_GET_ROTATED_VECTORS_ID | 31 |
| xl_position_on_orbit      | XL_POSITION_ON_ORBIT      | 32 |
| xl_quaternions_to_vectors | XL_QUATERNIONS_TO_VEC_ID  | 33 |
| xl_vectors_to_quaternions | XL_VEC_TO_QUATERNIONS_ID  | 34 |
| xl_star_catalog           | XL_STAR_CATALOG_ID        | 35 |
| xl_cart_to_radec          | XL_CART_TO_RADEC_ID       | 36 |
| xl_radec_to_cart          | XL_RADEC_TO_CART_ID       | 37 |
| xl_topocentric_to_ef      | XL_TOPOCENTRIC_TO_EF_ID   | 38 |
| xl_ef_to_topocentric      | XL_EF_TO_TOPOCENTRIC_ID   | 39 |
| Error Handling Functions  |                           |    |
| xl_verbose                | not applicable            |    |
| xl_silent                 |                           |    |
| xl_get_code               |                           |    |
| xl_get_msg                |                           |    |
| xl_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

The runtime performances of few of the CFI functions are improved to a large extent if they are called two consecutive times keeping constant some of their inputs:

- `xl_change_cart_cs`: time (reference and value).
- `xl_planet`: time (reference and value).
- `xl_star_radec`: time (reference and value).

In fact, the time, position, velocity and acceleration vectors do not need to keep exactly constant as long as the difference between two consecutive calls lays within the following thresholds:

- TAI/UT1/UTC time: 0.0864 microseconds
- GPS time: TBD microseconds

Furthermore, the same runtime improvement is achieved in other CFI functions that, although the user may not need to call two consecutive times with the same inputs, are called internally by other higher level CFI functions in those conditions, and thus improving the runtime performances of the latter. This is the case of the following CFI functions:

- `x1_cart_to_geod`: position, velocity and acceleration vectors
- `x1_sun`: time (reference and value)
- `x1_moon`: time (reference and value)

With the following thresholds:

- Position vector:  $0.6e-3$  m
- Velocity vector:  $0.6e-6$  m/s
- Acceleration vector:  $0.6e-9$  m/s<sup>2</sup>

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 (`XL_ERR_VECTOR_MAX_LENGTH`) as the maximum of all the Error Vector lengths. This value can therefore be safely used for every call of functions of this library.

## 6.2 General Enumerations

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

**Table 2: Enumerations within `EO_LIB` library**

| Input                 | Description                | Enumeration value                   | Long |
|-----------------------|----------------------------|-------------------------------------|------|
| Time reference        | Undefined                  | <code>XL_TIME_UNDEF</code>          | -1   |
|                       | TAI                        | <code>XL_TIME_TAI</code>            | 0    |
|                       | UTC                        | <code>XL_TIME_UTC</code>            | 1    |
|                       | UT1                        | <code>XL_TIME_UT1</code>            | 2    |
|                       | GPS                        | <code>XL_TIME_GPS</code>            | 3    |
| Processing format     | Standard                   | <code>XL_PROC</code>                | 0    |
| Transport time format | Standard                   | <code>XL_TRANS_STD</code>           | 0    |
|                       | Envisat Ground Segment     | <code>XL_TRANS_ENVI_GS</code>       | 11   |
|                       | CryoSat He by the Segment  | <code>XL_TRANS_CRYO_GS</code>       | 21   |
|                       | CryoSat General Telemetry  | <code>XL_TRANS_CRYO_TM</code>       | 22   |
|                       | CryoSat SIRAL Telemetry    | <code>XL_TRANS_CRYO_TM_SIRAL</code> | 23   |
|                       | SMOS transport time format | <code>XL_TRANS_SMOS_TM</code>       | 31   |

| Input   | Description  | Enumeration value                | Long |
|---|--|----------------------------------|------|
|   | GPS Second transport time format (number of seconds and microseconds elapsed since GPS epoch: 6 <sup>th</sup> January 1980)      | XL_TRANS_GENERIC_GPS_SEC         | 41   |
|   | GPS Week transport time format (number of weeks, seconds and microseconds elapsed since GPS epoch: 6 <sup>th</sup> January 1980) | XL_TRANS_GENERIC_GPS_WEEK        | 42   |
| ASCII time format                               | Undefined  | XL_ASCII_UNDEF                   | -1   |
|   | Standard   | XL_ASCII_STD                     | 11   |
|   | Standard with reference  | XL_ASCII_STD_REF                 | 12   |
|   | Standard with microseconds   | XL_ASCII_STD_MICROSEC            | 13   |
|   | Standard with reference and microseconds   | XL_ASCII_STD_REF_MICROSEC        | 14   |
|   | Compact  | XL_ASCII_COMPACT                 | 21   |
|   | Compact with reference   | XL_ASCII_COMPACT_REF             | 22   |
|   | Compact with microseconds  | XL_ASCII_COMPACT_MICROSEC        | 23   |
|   | Compact with reference and microseconds  | XL_ASCII_COMPACT_REF_MICROSEC    | 24   |
|   | Envisat  | XL_ASCII_ENVI                    | 31   |
|   | Envisat with reference   | XL_ASCII_ENVI_REF                | 32   |
|   | Envisat with microseconds  | XL_ASCII_ENVI_MICROSEC           | 33   |
|   | Envisat with reference and microseconds  | XL_ASCII_ENVI_REF_MICROSEC       | 34   |
|   | CCSDS-A  | XL_ASCII_CCSDSA                  | 41   |
|   | CCSDS-A with reference   | XL_ASCII_CCSDSA_REF              | 42   |
|   | CCSDS-A with microseconds  | XL_ASCII_CCSDSA_MICROSEC         | 43   |
|   | CCSDS-A with reference and microseconds  | XL_ASCII_CCSDSA_REF_MICROSEC     | 44   |
|   | CCSDS-A compact  | XL_ASCII_CCSDSA_COMPACT          | 51   |
|   | CCSDS-A compact with reference   | XL_ASCII_CCSDSA_COMPACT_REF      | 52   |
|   | CCSDS-A compact with microseconds  | XL_ASCII_CCSDSA_COMPACT_MICROSEC | 53   |
| CCSDS-A compact with reference and microseconds | XL_ASCII_CCSDSA_COMPACT_REF_MICROSEC   | 54                               |      |
| Time Initialization Mode                        | Initialization from file (data-driven)   | XL_SEL_FILE                      | 0    |
|   | Initialization within a time range   | XL_SEL_TIME                      | 1    |
|   | Initialization within a range of orbits (not used in LIB)  | XL_SEL_ORBIT                     | 2    |
|   |  | XL_SEL_DEFAULT                   | 3    |
| Time Initialization Model                       | Select the file type automatically   | XL_TIMEMOD_AUTO                  | -2   |
|   | User defined   | XL_TIMEMOD_USER                  | -1   |
|   | None   | XL_TIMEMOD_NONE                  | 0    |
|   | IERS Bulletin B - Table 1 (Predicted)  | XL_TIMEMOD_IERS_B_PREDICTED      | 1    |



| Input                     | Description   | Enumeration value                        | Long |
|---------------------------|---|--|------|
|                           | IERS Bulletin B - Table 2 (Restituted)                                      | XL_TIMEMOD_IERS_B_RESTITUTED             | 2    |
|                           | FOS Predicted Orbit File  | XL_TIMEMOD_FOS_PREDICTED                 | 3    |
|                           | FOS Restituted Orbit File   | XL_TIMEMOD_FOS_RESTITUTED                | 4    |
|                           | DORIS Preliminary Orbit   | XL_TIMEMOD_DORIS_PRELIMINARY             | 5    |
|                           | DORIS Precise Orbit   | XL_TIMEMOD_DORIS_PRECISE                 | 6    |
|                           | DORIS Navigator   | XL_TIMEMOD_DORIS_NAVIGATOR               | 7    |
|                           | Orbit Scenario File   | XL_TIMEMOD_OSF                           | 8    |
|                           | IERS Bulletin A – Prediction table  | XL_TIMEMOD_IERS_A_ONLY_PREDICTION        | 9    |
|                           | IERS Bulletin A – Prediction table and extrapolation formula                | XL_TIMEMOD_IERS_A_PREDICTION_AND_FORMULA | 10   |
|                           | IERS Bulletin B plus IERS Bulletin A (only prediction table for Bulletin A) | XL_TIMEMOD_IERS_B_AND_A_ONLY_PREDICTION  | 11   |
| Reference frame           | Barycentric Mean of 2000  | XL_BM2000                                | 1    |
|                           | Heliocentric Mean of 2000   | XL_HM2000                                | 2    |
|                           | Geocentric Mean of 2000   | XL_GM2000                                | 3    |
|                           | Mean of Date  | XL_MOD                                   | 4    |
|                           | True of Date  | XL_TOD                                   | 5    |
|                           | Earth Fixed   | XL_EF                                    | 6    |
|                           | Barycentric Mean of 1950  | XL_BM1950                                | 7    |
|                           | Galactic Coordinates  | XL_GALACTIC                              | 8    |
| Extended reference frames | Barycentric Mean of 2000.0  | BAR_MEAN_2000                            | 1    |
|                           | Heliocentric Mean of 2000.0   | HEL_MEAN_2000                            | 2    |
|                           | Geocentric Mean of 2000.0   | GEO_MEAN_2000                            | 3    |
|                           | Mean of date  | MEAN_DATE                                | 4    |
|                           | True of date  | TRUE_DATE                                | 5    |
|                           | Pseudo Earth Fixed  | PSEUDO_EARTH_FIXED                       | 6    |
|                           | Earth Fixed   | EARTH_FIXED                              | 7    |
|                           | Barycentric Mean of 1950  | BAR_MEAN_1950                            | 8    |
|                           | Galactic Coordinates  | GALACTIC                                 | 9    |
|                           | Satellite relative actual reference cs                                      | SAT_ACT_REF                              | 10   |
|                           | Quasi-Mean of Date  | QUASI_MEAN_DATE                          | 11   |
|                           | Pseudo-True of Date   | PSE_TRUE_DATE                            | 12   |
|                           | Topocentric coordinate system   | TOPOCENTRIC                              | 13   |
|                           | Satellite reference frame   | SAT_REF                                  | 14   |
|                           | Satellite relative reference frame  | SAT_REL_REF                              | 15   |
| Kepler OSV mode           | Mean Kepler State Vector  | XL_KEPLER_MEAN                           | 1    |
|                           | Osculating Kepler State Vector  | XL_KEPLER_OSC                            | 2    |
| Planet ID                 | Mercury   | XL_MERCURY                               | 1    |
|                           | Venus   | XL_VENUS                                 | 2    |

| Input                             | Description                             | Enumeration value          | Long |
|-----------------------------------|---|----------------------------|------|
|                                   | Earth-Moon barycenter                   | XL_EM_BAR                  | 3    |
|                                   | Mars                                    | XL_MARS                    | 4    |
|                                   | Jupiter                                 | XL_JUPITER                 | 5    |
|                                   | Saturn                                  | XL_SATURN                  | 6    |
|                                   | Uranus                                  | XL_URANUS                  | 7    |
|                                   | Neptune                                 | XL_NEPTUNE                 | 8    |
| Calculation mode                  | Position                                | XL_CALC_POS                | 1    |
|                                   | Position and velocity                   | XL_CALC_POS_VEL            | 2    |
|                                   | Position, velocity and acceleration     | XL_CALC_POS_VEL_ACC        | 3    |
| AOCS mode                         | Default Cx, Cy, Cz values               | XL_AOCS_DEFAULT            | 0    |
|                                   | User defined Cx, Cy, Cz values          | XL_AOCS_USER               | 1    |
|                                   | Geocentric pointing                     | XL_AOCS_GPM                | 2    |
|                                   | Local normal pointing                   | XL_AOCS_LNP                | 3    |
|                                   | Yaw steering + local normal pointing    | XL_AOCS_YSM                | 4    |
| Angle Type                        | True Latitude (TOD)                     | XL_ANGLE_TYPE_TRUE_LAT_TOD | 1    |
|                                   | Mean Latitude (TOD)                     | XL_ANGLE_TYPE_MEAN_LAT_TOD | 2    |
| Derivatives                       | No derivative                           | XL_NO_DER                  | 0    |
|                                   | First in his joy is also calculated     | XL_DER_1ST                 | 1    |
|                                   | First and second derivative.            | XL_DER_2ND                 | 2    |
| Type of <i>lds</i>                | Unknown                                 | XL_INIT_UNKNOWN            | 0    |
|                                   | <i>runId</i>                            | XL_INIT_RUN                | 1    |
|                                   | <i>timeId</i>                           | XL_INIT_TIME               | 2    |
|                                   | <i>orbitId</i> (not used in LIB)        | XO_INIT_ORBIT              | 3    |
|                                   | <i>propagId</i> (not used in LIB)       | XO_INIT_PROPAG             | 4    |
|                                   | <i>interpold</i> (not used in LIB)      | XO_INIT_INTERPOL           | 5    |
|                                   | <i>sat_nom_att_Id</i> (not used in LIB) | XP_INIT_SAT_NOM_ATT        | 6    |
|                                   | <i>sat_att_Id</i> (not used in LIB)     | XP_INIT_SAT_ATT            | 7    |
|                                   | <i>instr_att_Id</i> (not used in LIB)   | XP_INIT_INSTR_ATT          | 8    |
|                                   | <i>attitudeld</i> (not used in LIB)     | XP_INIT_ATTITUDE           | 9    |
|                                   | <i>atmosId</i> (not used in LIB)        | XP_INIT_ATMOS              | 10   |
|                                   | <i>demId</i> (not used in LIB)          | XP_INIT_DEM                | 11   |
| <i>targetId</i> (not used in LIB) | XP_INIT_TARGET                          | 12                         |      |
| Boolean values                    | False                                   | XL_FALSE                   | 0    |
|                                   | True                                    | XL_TRUE                    | 1    |
| Star Catalogues                   | FK4 Star catalogue                      | XL_FK4                     | 0    |
|                                   | FK5 Star catalogue                      | XL_FK5                     | 1    |
| Vector mode flag                  | Point location                          | XL_MODE_FLAG_LOCATION      | 0    |
|                                   | Direction vector                        | XL_MODE_FLAG_DIRECTION     | 1    |
| Model sets                        | CFI Default models                      | XL_MODEL_DEFAULT           | 0    |

| Input  | Description                            | Enumeration value           | Long |
|--|--|-----------------------------|------|
|  | User defined models                    | XL_MODEL_CONFIG             | 1    |
| Model types  | Earth model                            | XL_MODEL_TYPE_EARTH         | 0    |
|  | Sun model                              | XL_MODEL_TYPE_SUN           | 1    |
|  | Moon model                             | XL_MODEL_TYPE_MOON          | 2    |
|  | Planet model                           | XL_MODEL_TYPE_PLANET        | 3    |
|  | Star model                             | XL_MODEL_TYPE_STAR          | 4    |
|  | Nutation model                         | XL_MODEL_TYPE_NUTATION      | 5    |
|  | Precession model                       | XL_MODEL_TYPE_PRECESSION    | 6    |
|  | Constant model                         | XL_MODEL_TYPE_CONSTANTS     | 7    |
|  | Number of models                       | XL_NUM_MODEL_TYPES_ENUM     | 8    |
| Earth model  | Earth Default model                    | XL_MODEL_EARTH_DEFAULT      | 0    |
| Sun model  | Sun Default model                      | XL_MODEL_SUN_DEFAULT        | 0    |
| Moon model   | Moon Default model                     | XL_MODEL_MOON_DEFAULT       | 0    |
| Planet model   | Planet Default model                   | XL_MODEL_PLANET_DEFAULT     | 0    |
| Star model   | Star Default model                     | XL_MODEL_STAR_DEFAULT       | 0    |
| Nutation model   | Nutation Default model                 | XL_MODEL_NUTATION_DEFAULT   | 0    |
| Precession model   | Precession Default model               | XL_MODEL_PRECESSION_DEFAULT | 0    |
| Constants model  | Contants Default model                 | XL_MODEL_CONSTANTS_DEFAULT  | 0    |
| Bulletin type  | File is not an IERS Bulletin           | XL_NO_BULLETIN              | 0    |
|  | IERS Bulletin B                        | XL_BULLETIN_B               | 1    |
|  | IERS Bulletin A                        | XL_BULLETIN_A               | 2    |
|  | IERS Bulletin B plus IERS Bulletin A   | XL_BULLETIN_B_AND_A         | 3    |
| Extrapolation formulas activation                        | IERS A Extrapolation formulas enabled  | XL_FORMULA_ENABLED          | 0    |
|  | IERS A Extrapolation formulas disabled | XL_FORMULA_DISABLED         | 1    |
| Data origin for time initialization with xl_time_id_init | Data read from file                    | XL_FILE_DATA                | 0    |
|  | Data from user time correlation data   | XL_TIME_CORRELATIONS_DATA   | 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**.

The meanings and units of the different array elements from the Transport time strongly depend upon the selected Transport format (by means of the Transport format ID). The table below shows the choices:

**Table 3: Transport time formats**

| Input                     | Array Element | Unit and shun)         | Allowed Range   |
|---------------------------|---------------|------------------------|---|
| XL_TRANS_STD              | [0]           | Integer days           | [-18262,36524]  |
|                           | [1]           | Integer seconds        | [0,86399]   |
|                           | [2]           | Integer microseconds   | [0,999999]  |
| XL_TRANS_ENVI_GS          | [0]           | Integer days           | [-18262,36524]  |
|                           | [1]           | Integer seconds        | [0,86399]   |
|                           | [2]           | Integer microseconds   | [0,999999]  |
| XL_TRANS_CRYO_GS          | [0]           | Integer days           | [-18262,36524]  |
|                           | [1]           | Integer seconds        | [0,86399]   |
|                           | [2]           | Integer microseconds   | [0,999999]  |
| XL_TRANS_CRYO_TM          | [0]           | Integer days           | [-18262,36524]  |
|                           | [1]           | Integer milliseconds   | [0,86399999]  |
|                           | [2]           | Integer microseconds   | [0,999]   |
| XL_TRANS_CRYO_TM_SIRAL    | [0]           | Integer days           | [-18262,36524]  |
|                           | [1]           | Integer milliseconds   | [0,86399999]  |
|                           | [2]           | Integer microseconds   | [0,999]   |
|                           | [3]           | SIRAL extra counter    | [0,1745454545]  |
| XL_TRANS_SMOS_TM          | [0]           | Week number            | [-1566, 6260]   |
|                           | [1]           | Seconds of week        | [0, 604799]   |
|                           | [2]           | Fraction of seconds    | [0, 65535]  |
| XL_TRANS_GENERIC_GPS_SEC  | [0]           | Number of seconds      | [0, INT_MAX]<br>Note: INT_MAX is a macro that returns the maximum number a C int variable can store |
|                           | [1]           | Number of microseconds | [0, 999999]   |
| XL_TRANS_GENERIC_GPS_WEEK | [0]           | Number of weeks        | [0, 6260]   |
|                           | [1]           | Number of seconds      | [0, 604799]   |
|                           | [2]           | Number of microseconds | [0, 999999]   |

The string characteristics of the ASCII time formats depends strongly upon the selected ASCII format (by means of the ASCII format ID). The tables below show the available choices:

Note that the value of 86400 for seconds (and 86400000 for milliseconds) is accepted only for UTC in case a leap second is being introduced. This may happen only at 23:59 minutes and only on four days of the year (31/03, 30/06, 30/09, 31/12). The decision to introduce a leap second in UTC is the responsibility of the International Earth Rotation Service (IERS). See [IERS] for further details.

For further details on the SIRAL extra counter for the Cryosat mission please see [MCD].

**Table 4: Basic ASCII time formats**

| Input                   | String format          |
|-------------------------|------------------------|
| XL_ASCII_UNDEF          | -                      |
| XL_ASCII_STD            | "yyyy-mm-dd_hh:nn:ss"  |
| XL_ASCII_COMPACT        | "yyyymmdd_hhnnss"      |
| XL_ASCII_ENVI           | "dd-mmm-yyyy hh:nn:ss" |
| XL_ASCII_CCSDSA         | "yyyy-mm-ddThh:nn:ss"  |
| XL_ASCII_CCSDSA_COMPACT | "yyyymmddThhnnss"      |

**Table 5: Derived ASCII time formats**

| Input                                | String format                     |
|--------------------------------------|-----------------------------------|
| XL_ASCII_STD_REF                     | "RRR=yyyy-mm-dd_hh:nn:ss"         |
| XL_ASCII_STD_MICROSEC                | "yyyy-mm-dd_hh:nn:ss.uuuuuu"      |
| XL_ASCII_STD_REF_MICROSEC            | "RRR=yyyy-mm-dd_hh:nn:ss.uuuuuu"  |
| XL_ASCII_COMPACT_REF                 | "RRR=yyyymmdd_hhnnss"             |
| XL_ASCII_COMPACT_MICROSEC            | "yyyymmdd_hhnnssuuuuuu"           |
| XL_ASCII_COMPACT_REF_MICROSEC        | "RRR=yyyymmdd_hhnnssuuuuuu"       |
| XL_ASCII_ENVI_REF                    | "RRR=dd-mmm-yyyy hh:nn:ss"        |
| XL_ASCII_ENVI_MICROSEC               | "dd-mmm-yyyy hh:nn:ss.uuuuuu"     |
| XL_ASCII_ENVI_REF_MICROSEC           | "RRR=dd-mmm-yyyy hh:nn:ss.uuuuuu" |
| XL_ASCII_CCSDSA_REF                  | "RRR=yyyy-mm-ddThh:nn:ss"         |
| XL_ASCII_CCSDSA_MICROSEC             | "yyyy-mm-ddThh:nn:ss.uuuuuu"      |
| XL_ASCII_CCSDSA_REF_MICROSEC         | "RRR=yyyy-mm-ddThh:nn:ss.uuuuuu"  |
| XL_ASCII_CCSDSA_COMPACT_REF          | "RRR=yyyymmddThhnnss"             |
| XL_ASCII_CCSDSA_COMPACT_MICROSEC     | "yyyymmddThhnnssuuuuuu"           |
| XL_ASCII_CCSDSA_COMPACT_REF_MICROSEC | "RRR=yyyymmddThhnnssuuuuuu"       |

where:

- *yyyy* stands for the year
- *mm* stands for the month expressed as a numerical count, i.e. 01 for January, etc
- *mmm* stands for the month expressed in abbreviations, i.e. JAN, MAR, etc
- *dd* stands for the day of month
- *ddd* stands for the day of the year
- *hh* stands for the hour in the day
- *nn* stands for the minutes within a hour

- *ss* stands for the seconds within a minute
- *uuuuuu* stands for the microseconds within a second
- *RRR* stands for the time reference (TAI, UTC, UT1 or GPS)

In ASCII formats two values are defined, by convention, as Beginning of Mission (BOM) and End of Mission (EOM). These values are listed, for the various ASCII time formats, in Table 6 and Table 7.

Usually a date with all zeros is seen as EOM, and a date with all nines is considered EOM. The only exception are the ENVISAT-specific formats, which use as EOM the date December 31st, 2078 at 23:59:59.999999.

Format transformations of BOM and EOM between ASCII format is allowed.

Time reference is not considered in BOM or EOM, thus any time reference is accepted (TAI, UTC, UT1 or GPS) for the values in Table 6 and Table 7..

BOM and EOM do not have an equivalent in Processing or Transport formats, so if the user tries to convert them from ASCII to another non-ASCII format an error will occur.

**Table 6: Definition of BOM and EOM for basic ASCII time formats**

| ASCII format            | Beginning of Mission   | End of Mission         |
|-------------------------|------------------------|------------------------|
| XL_ASCII_UNDEF          | -                      | -                      |
| XL_ASCII_STD            | "0000-00-00_00:00:00"  | "9999-99-99_99:99:99"  |
| XL_ASCII_COMPACT        | "00000000_000000"      | "99999999_999999"      |
| XL_ASCII_ENVI           | "00-000-0000_00:00:00" | "31-DEC-2078 23:59:59" |
| XL_ASCII_CCSDSA         | "0000-00-00T00:00:00"  | "9999-99-99T99:99:99"  |
| XL_ASCII_CCSDSA_COMPACT | "00000000T000000"      | "99999999T999999"      |

**Table 7: Definition of BOM and EOM for derived ASCII time formats**

| ASCII format                  | Beginning of Mission              | End of Mission                    |
|-------------------------------|-----------------------------------|-----------------------------------|
| XL_ASCII_STD_REF              | "RRR=0000-00-00_00:00:00"         | "RRR=9999-99-99_99:99:99"         |
| XL_ASCII_STD_MICROSEC         | "0000-00-00_00:00:00.000000"      | "9999-99-99_99:99:99.999999"      |
| XL_ASCII_STD_REF_MICROSEC     | "RRR=0000-00-00_00:00:00.000000"  | "RRR=9999-99-99_99:99:99.999999"  |
| XL_ASCII_COMPACT_REF          | "RRR=00000000_000000"             | "RRR=99999999_999999"             |
| XL_ASCII_COMPACT_MICROSEC     | "00000000_000000000000"           | "99999999_999999999999"           |
| XL_ASCII_COMPACT_REF_MICROSEC | "RRR=00000000_000000000000"       | "RRR=99999999_999999999999"       |
| XL_ASCII_ENVI_REF             | "RRR=00-000-0000_00:00:00"        | "RRR=31-DEC-2078 23:59:59"        |
| XL_ASCII_ENVI_MICROSEC        | "00-000-0000_00:00:00.000000"     | "31-DEC-2078 23:59:59.999999"     |
| XL_ASCII_ENVI_REF_MICROSEC    | "RRR=00-000-0000_00:00:00.000000" | "RRR=31-DEC-2078 23:59:59.999999" |
| XL_ASCII_CCSDSA_REF           | "RRR=0000-00-00T00:00:00"         | "RRR=9999-99-99T99:99:99"         |
| XL_ASCII_CCSDSA_MICROSEC      | "0000-00-00T00:00:00.000000"      | "9999-99-99T99:99:99.999999"      |

| ASCII format                     | Beginning of Mission             | End of Mission                   |
|----------------------------------|----------------------------------|----------------------------------|
| XL_ASCII_CCSDSA_REF_MICROSEC     | "RRR=0000-00-00T00:00:00.000000" | "RRR=9999-99-99T99:99:99.999999" |
| XL_ASCII_CCSDSA_COMPACT_REF      | "RRR=00000000T000000"            | "RRR=99999999T999999"            |
| XL_ASCII_CCSDSA_COMPACT_MICROSEC | "00000000T000000000000"          | "99999999T999999999999"          |

where:

- *RRR* stands for the time reference (TAI, UTC, UT1 or GPS)

### 6.3 Data Structures

The aim of the current section is to present the data structures that are used in the EO\_LIB library. The structures are currently used for the CFI Identifiers accessor functions. The following table show the structures with their names and the data that contain:

**Table 8: EO\_LIB structures**

| Structure name | Data          |               |   |
|----------------|---------------|---------------|---|
|                | Variable Name | C type        | Description   |
| xl_par_der     | deriv         | XL_Deriv_enum | Flag to indicate if the 1st and 2nd derivatives are defined             |
|                | p             | double        | The parameter, expressed in the appropriate units                       |
|                | pd            | double        | 1st time derivative of the parameter                                    |
|                | p2d           | double        | 2nd time derivative of the parameter                                    |
| xl_cord        | cs            | XL_CS_rl_enum | Coordinate reference frame  |
|                | deriv         | XL_Deriv_enum | Flag to indicate if the 1st and 2 <sup>nd</sup> derivatives are defined |
|                | v             | double [3]    | Vector  |
|                | vd            | double [3]    | Vector rate   |
|                | v2d           | double [3]    | Vector rate-rate  |

| Structure name | Data                 |                     |  |
|----------------|----------------------|---------------------|--|
|                | Variable Name        | C type              | Description  |
| xl_par_der     | deriv                | XL_Deriv_enum       | Flag to indicate if the 1st and 2nd derivatives are defined            |
|                | p                    | double              | The parameter, expressed in the appropriate units                      |
|                | pd                   | double              | 1st time derivative of the parameter                                   |
|                | p2d                  | double              | 2nd time derivative of the parameter                                   |
| xl_cs_tra      | azel_flag            | XL_Boolean          | Flag to indicate if an azimuth/elevation definition has been provided. |
|                | azel_def             | xl_az_el_definition | Azimuth/elevation definition   |
|                | ref_i                | XL_CS_rl_enum       | Initial reference frame  |
|                | ref_f                | XL_Attitude_fr_enum | final reference frame  |
|                | amb_flag             | XL_Boolean          | Ambiguity flag   |
|                | deriv                | XL_Deriv_enum       | Flag to indicate if the 1st and 2nd derivatives are defined            |
|                | v                    | double [3]          | Translation vector from ref_i to ref_f                                 |
|                | vd                   | double [3]          | Translation rate vector from ref_i to ref_f                            |
|                | v2d                  | double [3]          | Translation rate-rate vector from ref_i to ref_f                       |
|                | m                    | double [3][3]       | Rotation matrix from ref_i to ref_f                                    |
|                | md                   | double [3][3]       | Rotation matrix rate from ref_i to ref_f                               |
|                | m2d                  | double [3][3]       | Rotation matrix rate-rate from ref_i to ref_f                          |
|                | xl_time_correlations | tai_time            | double   |
| ut1_time       |                      | double              | UT1 time   |
| tai_utc        |                      | double              | difference between TAI and UTC   |
| tai_ut1        |                      | double              | difference between TAI and UT1   |
| tai_gps        |                      | double              | difference between TAI and GPS   |
| xl_leap_second | flag                 | long                | XL_TRUE if the leap second exists                                      |
|                | utc_time             | double              | UTC time for the leap second   |



| Structure name      | Data                     |                             |  |
|---------------------|--------------------------|-----------------------------|--|
|                     | Variable Name            | C type                      | Description  |
| xl_par_der          | deriv                    | XL_Deriv_enum               | Flag to indicate if the 1st and 2nd derivatives are defined  |
|                     | p                        | double                      | The parameter, expressed in the appropriate units  |
|                     | pd                       | double                      | 1st time derivative of the parameter   |
|                     | p2d                      | double                      | 2nd time derivative of the parameter   |
| xl_time_id_data     | iers_bulletin_type       | long                        | Bulletin type (see Bulletin type)  |
|                     | iers_formula_flag        | long                        | IERS extrapolation formula enabled or disabled (see Extrapolation formulas activation)               |
|                     | prediction_first_record  | long                        | Indicates the 1st record belonging to bulletin A if applicable (starting at 0).                      |
|                     | polar_motion_formula     | xl_polar_motion_formula     | Polar motion formula parameters  |
|                     | time_correlation_formula | xl_time_correlation_formula | Time correlation formula parameters  |
|                     | num_lines                | long                        | Number of records in the array with the time correlations  |
|                     | time_str                 | xl_time_correlations*       | Array with the time correlations   |
|                     | polar_motion_params      | xl_polar_motion_params*     | Array with the polar motion parameters (it can be NULL if there are no parameters in initialization) |
|                     | leap_sec                 | xl_leap_second              | Leapsecond information   |
| xl_az_el_definition | az_0_axis                | long                        | Azimuth 0deg axis (one of the values in XL_Axis_enum)  |
|                     | az_90_axis               | long                        | Azimuth 90deg axis (one of the values in XL_Axis_enum)   |
|                     | el_90_axis               | long                        | Elevation 90deg axis (one of the values in XL_Axis_enum)   |

| Structure name | Data             |  |  |
|----------------|------------------|--|--|
|                | Variable Name    | C type   | Description  |
| xl_par_der     | deriv            | XL_Deriv_enum  | Flag to indicate if the 1st and 2nd derivatives are defined        |
|                | p                | double   | The parameter, expressed in the appropriate units                  |
|                | pd               | double   | 1st time derivative of the parameter                               |
|                | p2d              | double   | 2nd time derivative of the parameter                               |
| xl_model_data  | earth_model      | long   | Earth model  |
|                | sun_model        | long   | Sun model  |
|                | moon_model       | long   | Moon model   |
|                | planet_model     | long   | Planets model  |
|                | star_model       | long   | Stars model  |
|                | nutaton_model    | long   | Nutation model   |
|                | precession_model | long   | Precession model   |
|                | constants_model  | long   | Constants model  |
|                | re               | double   | Earth equatorial radius [m]  |
|                | mu               | double   | Earth's gravitational constant [m <sup>3</sup> /s <sup>2</sup> ]   |
|                | j2               | double   | Second zonal harmonic  |
|                | j3               | double   | Third zonal harmonic   |
|                | j4               | double   | Fourth zonal harmonic  |
|                | major_axis       | double   | Semi-major axis [m]  |
|                | minor_axis       | double   | Semi-minor axis [m]  |
|                | ecc              | double   | First eccentricity [-]   |
|                | flat             | double   | Flattening [-]   |
|                | gcoef_0          | double   | Greenwich sidereal angle for t=0 (MJD 2000)                        |
|                | gcoef_1          | double   | 1st. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000) |
|                | gcoef_2          | double   | 2nd. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000) |
|                | gcoef_sim_0      | double   | Greenwich sidereal angle for t=0 (MJD 2000) (Simplified model)     |
| gcoef_sim_1    | double           | 1st. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000) (Simplified model)              |  |
| gcoef_sim_2    | double           | 2 <sup>nd</sup> . Derivative of the Greenwich sidereal angle for t=0 (MJD 2000) (Simplified model) |  |
| au             | double           | Astronomical units in kms  |  |

| Structure name              | Data              |                                 |   |
|-----------------------------|-------------------|---------------------------------|---|
|                             | Variable Name     | C type                          | Description   |
| xl_par_der                  | deriv             | XL_Deriv_enum                   | Flag to indicate if the 1st and 2nd derivatives are defined                               |
|                             | p                 | double                          | The parameter, expressed in the appropriate units   |
|                             | pd                | double                          | 1st time derivative of the parameter  |
|                             | p2d               | double                          | 2nd time derivative of the parameter  |
| xl_polar_motion_parameters  | x                 | double                          | x-axis is in the direction of the IERS Reference Meridian (IRM)                           |
|                             | y                 | double                          | y-axis is in the direction 90 degrees West longitude                                      |
| xl_polar_motion_formula     | ax                | double                          | x parameter formula: constant term  |
|                             | bx                | double                          | x parameter formula: cos(A) coefficient   |
|                             | cx                | double                          | x parameter formula: sin(A) coefficient   |
|                             | dx                | double                          | x parameter formula: cos(C) coefficient   |
|                             | ex                | double                          | x parameter formula: sin(C) coefficient   |
|                             | ay                | double                          | y parameter formula: constant term  |
|                             | by                | double                          | y parameter formula: cos(A) coefficient   |
|                             | cy                | double                          | y parameter formula: sin(A) coefficient   |
|                             | dy                | double                          | y parameter formula: cos(C) coefficient   |
|                             | ey                | double                          | y parameter formula: sin(C) coefficient   |
|                             | A_ref             | double                          | Reference day for A parameter formula (MJD2000)   |
|                             | A_div             | double                          | Divisor for A parameter formula   |
|                             | C_ref             | double                          | Reference day for C parameter formula (MJD2000)   |
| C_div                       | double            | Divisor for C parameter formula |   |
| xl_time_correlation_formula | a                 | double                          | Constant parameter  |
|                             | b                 | double                          | Linear coefficient  |
|                             | b_ref             | double                          | Reference day (MJD2000)   |
| xl_time_id_init_data_union  | file_set          | xd_eocfi_file_set               | Set of data from files  |
|                             | time_id_data      | xl_time_id_data                 | Data for time correlation initialization  |
| xl_time_id_init_data        | data_type         | long                            | Enumeration value from xl_time_data_origin_enum: XL_FILE_DATA or XL_TIME_CORRELATION_DATA |
|                             | time_id_init_data | xl_time_id_init_data_union      | Data for time initialization  |

## 7 CFI FUNCTIONS DESCRIPTION

The following sections describe each CFI function.

The calling interfaces are described for C users.

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 xl\_time\_ref\_init\_file

### 7.1.1 Overview

The `xl_time_ref_init_file` CFI function initializes time correlations between TAI, UTC, UT1 and GPS times from reference data files. The correlations provided by the different input files can be found in the following table.

**Table 9: Time reference correlations from reference files**

|                                      | TAI | UTC | UT1 | GPS | orbit |
|--------------------------------------|-----|-----|-----|-----|-------|
| FOS Predicted Orbit File             | X   | X   | X   | (x) | X     |
| FOS Restituted Orbit File            | X   | X   | X   | (x) | X     |
| DORIS Preliminary Orbit              | X   | X   | X   | (x) | X     |
| DORIS Precise Orbit                  | X   | X   | X   | (x) | X     |
| DORIS Navigator File                 | X   | X   | X   | (x) | X     |
| IERS Bulletin B format 1980          | X   | X   | X   | (x) |       |
| IERS Bulletin B format 2010          | X   | X   | X   | (x) |       |
| Orbit Scenario File                  | X   | X   | X   | (x) |       |
| IERS Bulletin A                      | X   | X   | X   | (X) |       |
| IERS Bulletin B plus IERS Bulletin A | X   | X   | X   | (X) |       |

Normally a single Predicted or DORIS Orbit file is sufficient to have all correlations needed (the (x) mark indicates that the GPS time correlation, although is not present within the file, can be simulated since it is always a fixed delta from TAI). The last updated IERS Bulletins can be downloaded from IERS bulletins web page ([IERS]).

When using an Orbit Scenario File, it must be taken into account that, since one orbital change can be far away from the following one, leap seconds could be calculated wrongly if there is more than one of the four possible leap second insertion points (end of March, end of June, end of September and end of December) between them.

When using a Bulletin B and a Bulletin A for initialization (B plus A initialization) it must be taken into account that:

- The first file in the input list must be Bulletin B, not Bulletin A.
- For Bulletin B, FINAL and PRELIMINARY tables are used; for Bulletin A, PREDICTION table is used.
- First record of Bulletin B must be before first record of Bulletin A.
- Last record of Bulletin A must be after last record of Bulletin B.
- In case of partial overlap (last record of Bulletin B is after first record of Bulletin A), the records of Bulletin A loaded are the ones that are after the last record of Bulletin B).
- In case of no overlap (first record of Bulletin B is after last record of Bulletin A), the gap must be less than 1 month (which is the periodicity of Bulletin B).

All other input files are ESA-provided. These initialization files could even be generated by the users by means of `EO_FILE_HANDLING` and `EO_DATA_HANDLING` CFI libraries.

In case multiple files are used for the time correlations initializations, the files should be time ordered. If there is overlap between files, the newest data have precedence.

For Orbit Scenario File, only one file is admitted. If more files are introduced a warning is raised and the computations are performed only with the first OSF introduced.

A complete calling sequence of the time reference computations is presented in section 4.2.

The validity interval of the initialization depends on the input file, according to the following table:

**Table 10: Initialization validity depending on input file**

|   | Validity start   | Validity stop   |
|---|--|---|
| FOS Predicted Orbit File<br>FOS Restituted Orbit File<br>DORIS Preliminary Orbit<br>DORIS Precise Orbit<br>DORIS Navigator File | Time of the first state vector in input files that belongs to input range  | Time of the last state vector in input files that belongs to input range  |
| ERS Bulletin B format 1980<br>IERS Bulletin B format 2010   | Time of first record in tables FINAL or PRELIMINARY (for PREDICTED initialization) or table SMOOTHED (for RESTITUTED initialization) that belongs to input range | Time of last record in tables FINAL or PRELIMINARY (for PREDICTED initialization) or table SMOOTHED (for RESTITUTED initialization) that belongs to input range |
| IERS Bulletin A (only prediction)   | Time of first record in PREDICTION table that belongs to input range   | Time of last record in PREDICTION table that belongs to input range   |
| IERS Bulletin A (prediction and formula)  | Time of first record in PREDICTION table that belongs to input range   | End of Mission  |
| IERS Bulletin B plus IERS Bulletin A  | Time of first record in FINAL or PRELIMINARY table of Bulletin B that belongs to input range   | Time of last record in PREDICTION table of Bulletin A that belongs to input range   |
| Orbit Scenario File   | Time of first orbital change in file   | End of mission  |

If any time operation is done with a time outside validity interval, the behaviour of the functions is the following:

- If the time is lower than the start validity time, a warning is returned and the time correlation used for the time computations is that of the first record stored in xl\_time\_id in the initialization.
- If the time is greater than the stop validity time, a warning is returned and the time correlation used for the time computations is that of the last record stored in xl\_time\_id in the initialization, except for the case of IERS Bulletin A prediction plus formula initialization, where the extrapolation formulas are used.

### 7.1.2 Calling interface

The calling interface of the xl\_time\_ref\_init\_file CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long time_model, n_files, time_init_mode, time_ref;
    long orbit0, orbit1;
    char **time_file;
    double time0, time1, val_time0, val_time1;
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_TIME_REF_INIT_FILE], status;

    status = xl_time_ref_init_file (&time_model, &n_files,
                                   time_file, &time_init_mode,
                                   &time_ref, &time0, &time1,
                                   &orbit0, &orbit1,
                                   &val_time0, &val_time1,
                                   &time_id, ierr);
}

```

### 7.1.3 Input parameters

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

*Table 11: Input parameters of xl\_time\_ref\_init\_file function*

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

|                |         |  |  |                                  |  |
|----------------|---------|--|--|----------------------------------|--|
| time_model     | long *  |  | Time model ID  |                                  | Complete except XL_TIMEMOD_USER<br><u>Note:</u> When the input file is an IERS Bulletin B format 1980 file and the time mode is XL_TIMEMOD_AUTO, then the time model is set automatically to XL_TIMEMOD_IERS_B_RESTITUTED<br><u>Note:</u> When the input file is an IERS Bulletin B format 2010, the time models XL_TIMEMOD_IERS_B_RESTITUTED and XL_TIMEMOD_IERS_B_PREDICTED coincide.<br><u>Note:</u> Whe the input file is an IERS Bulletin A and the time mode is XL_TIMEMOD_AUTO, the the time model is set automatically to XL_TIMEMOD_IERS_A_PREDICTION_AND_FORMULA |
| n_files        | long *  |  | Number of reference data files   |                                  | > 0  |
| time_file      | char**  |  | Filenames of the reference data files  |                                  |  |
| time_init_mode | long *  |  | Flag for selecting the time range of the initialisation: It could be the whole file (XL_SEL_FILE), the orbit range given by orbit0-orbit1 (XO_SEL_ORBIT) or the time range given by time0-time1(XO_SEL_TIME) |                                  | Select either:<br>· XL_SEL_FILE<br>· XL_SEL_ORBIT<br>· XL_SEL_TIME<br><br>- XL_SEL_ORBIT is not allowed for IERS Bulletins (any format) nor DORIS Navigator files<br>- XL_SEL_ORBIT and XL_SEL_TIME are not enabled for OSF  |
| time_ref       | long *  |  | Time reference ID  |                                  | Complete.<br>If the input file is a DORIS Navigator file and the time_init_mode is XL_SEL_TIME, then only time_ref allowed is XL_TIME_UTC.   |
| time0          | double* |  | If: time_init_mode=XL_SEL_TIME Start of the time range defined by [time0,time1]  | Decimal days (Processing format) | [-18262.0,36524.0]   |
| time1          | double* |  | If: time_init_mode=XL_SEL_TIME   | Decimal days (Processing format) | [-18262.0,36524.0] > time0   |



|        |       |   |   |         |         |
|--------|-------|---|---|---------|---------|
|        |       |   | End of the time range defined by [time0,time1]  | format) |         |
| orbit0 | long* | - | If:<br>time_init_mode=XL_SEL_ORBIT Absolute orbit number corresponding to the start of the time range defined by [ANX <sub>orbit0</sub> , ANX <sub>orbit1+1</sub> ] | -       | >= 0    |
| orbit1 | long* | - | If:<br>time_init_mode=XL_SEL_ORBIT Absolute orbit number corresponding to the end of the time range defined by [ANX <sub>orbit0</sub> , ANX <sub>orbit1+1</sub> ]   | -       | >orbit0 |

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.

### 7.1.4 Output parameters

The output parameters of the xl\_time\_ref\_init\_file CFI function are:

Table 12: Output parameters of xl\_time\_ref\_init\_file function

| C name                | C type      | Array Element | Description (Reference)                                   | Unit (Format)                    | Allowed Range      |
|-----------------------|-------------|---------------|---|----------------------------------|--------------------|
| xl_time_ref_init_file | long        | -             | Status flag   | -                                | -                  |
| val_time0             | double*     | -             | Validity start time of the initialization (see table 10). | Decimal days (Processing format) | [-18262.0,36524.0] |
| val_time1             | double*     | -             | Validity end time of the initialization (see table 10).   | Decimal days (Processing format) | [-18262.0,36524.0] |
| time_id               | xl_time_id* | -             | Structure that contains the time correlations.            | -                                | -                  |
| ierr                  | long        | -             | Error vector  | -                                | -                  |

Note that *val\_time0* and *val\_time1* can define a validity range different to that requested by the user. This range gives the maximum coverage provided by the input files within the margins selected by the user (see table 10).

It has to be remarked that if the input time is outside the range of initialization, transformations are performed anyway, using the closest correlation data (or the extrapolation formula for IERS Bulletin A

prediction plus formula initialization). However a warning is returned, since there is no guarantee that the correlation is correct.

### 7.1.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_time_ref_init_file` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_time_ref_init_file` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 13: Error messages of `xl_time_ref_init_file` function**

| Error type | Error message                                | Cause and impact         | Error code                                       | Error No |
|------------|--|--------------------------|--|----------|
| ERR        | Time model ID is not correct                 | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_TIME_MODEL_ERR         | 0        |
| ERR        | Non-positive number of data files            | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_N_FILES_ERR            | 1        |
| ERR        | Incorrect file names                         | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_FILE_NAMES_ERR         | 2        |
| ERR        | Time init mode ID is not correct             | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_INIT_MODE_ERR          | 3        |
| ERR        | Time reference ID is not correct             | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_TIME_ERR               | 4        |
| ERR        | Reference start time out of limits           | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_DAY_0_ERR              | 5        |
| ERR        | Reference end time out of limits             | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_DAY_1_ERR              | 6        |
| ERR        | Wrong reference time range                   | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_DAY_RANGE_ERR          | 7        |
| ERR        | Reference start orbit is negative            | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_ORB_0_ERR              | 8        |
| ERR        | Reference end orbit is negative              | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_ORB_1_ERR              | 9        |
| ERR        | Wrong reference orbit range                  | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_ORB_RANGE_ERR          | 10       |
| ERR        | File does not exist                          | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_FILE_ERR               | 11       |
| ERR        | Time table is empty or has wrong format      | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_TABLE_ERR              | 12       |
| ERR        | Time range from file is outside input range  | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_TIME_OUTSIDE_RANGE_ERR | 13       |
| ERR        | Orbit range from file is outside input range | No calculation performed | XL_CFI_TIME_REF_INIT_FILE_ORB_OUTSIDE_RANGE_ERR  | 14       |

|      |   |   |  |    |
|------|---|---|--|----|
| ERR  | Memory allocation error   | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_MEMORY_ERR             | 15 |
| ERR  | Error in reading file   | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_READ_FILE_ERR          | 16 |
| ERR  | Time reference ID is already initialized  | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_STATUS_ERR             | 17 |
| ERR  | Could not find out the input file types   | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_DETECT_FILE_ERR        | 18 |
| ERR  | The input file type is not correct  | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_WRONG_FILE_TYPE_ERR    | 19 |
| ERR  | Input time reference should be UTC for DORIS Navigator files  | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_TIME_REF_FOR_DORIS_ERR | 20 |
| ERR  | Error loading orbit files   | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_LOAD_FILES_ERR         | 21 |
| WARN | Only one OSF file is admitted for this initialisation mode  | Calculation performed using first OSV file introduced | XL_CFI_TIME_REF_INIT_FILE_ONLY_FIRST_OSF_WARN    | 22 |
| WARN | Time init mode option not currently enabled for file  | Calculation performed with option XL_SEL_FILE         | XL_CFI_TIME_REF_INIT_FILE_INIT_MODE_WARN         | 23 |
| ERR  | Input IERS Bulletins and initialization mode incompatible   | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_IERS_INIT_ERR          | 24 |
| ERR  | Input Bulletins B and A are not compatible. The gap or the overlap between the 2 files is not correct | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_IERS_B_A_WRONG_ERR     | 25 |
| ERR  | Could not compute leap second   | No calculation performed                              | XL_CFI_TIME_REF_INIT_FILE_LEAP_SEC_ERR           | 26 |

## 7.2 xl\_time\_ref\_init

### 7.2.1 Overview

The `xl_time_ref_init` CFI function initializes time correlations between TAI, UTC, UT1 and GPS times from input reference times for time ranges from -18262.0 and +36524.0 decimal days.

A complete calling sequence of the time reference computations is presented in section 4.2.

### 7.2.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long orbit_num;
    double time[4], anx_time, orbit_duration;
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_TIME_REF_INIT], status;

    status = xl_time_ref_init (time, &orbit_num, &anx_time,
                             &orbit_duration, &time_id, ierr);
}
```

Note that input time vector must be indexed using the existing enumeration for time references.

The `XL_NUM_ERR_TIME_REF_INIT` constant is defined in the file `explorer_lib.h`.

### 7.2.3 Input parameters

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

**Table 14: Input parameters of `xl_time_ref_init` function**

| C name | C type    | Array Element | Description (Reference) | Unit (Format)                       | Allowed Range      |
|--------|-----------|---------------|-------------------------|-------------------------------------|--------------------|
| time   | double[4] | [0]           | TAI input time          | Decimal days<br>(Processing format) | [-18262.0,36524.0] |
|        |           | [1]           | UTC input time          | Decimal days<br>(Processing format) | [-18262.0,36524.0] |
|        |           | [2]           | UT1 input time          | Decimal days<br>(Processing format) | [-18262.0,36524.0] |
|        |           | [3]           | GPS input time          | Decimal days<br>(Processing format) | [-18262.0,36524.0] |

|                |         |   |  |         |                    |
|----------------|---------|---|--|---------|--------------------|
| orbit_num      | long*   | - | Absolute orbit number at the reference time              | -       | >=0                |
| anx_time       | double* | - | Time since Ascending node crossing at the reference time | Seconds | [0,orbit_duration] |
| orbit_duration | double* | - | Duration of the orbit containing the reference time      | Seconds | >0                 |

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

- Time vector can be accessible by means of enumeration values, as defined in [GEN\_SUM].

### 7.2.4 Output parameters

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

**Table 15: Output parameters of `xl_time_ref_init` function**

| C name                        | C type                   | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|-------------------------------|--------------------------|---------------|--|---------------|---------------|
| <code>xl_time_ref_init</code> | long                     | -             | Status flag                                    | -             | -             |
| <code>time_id</code>          | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -             | -             |
| <code>ierr</code>             | long                     | -             | Error vector                                   | -             | -             |

### 7.2.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_time_ref_init` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_ref_init` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 16: Error messages of `xl_time_ref_init` function**

| Error type | Error message            | Cause and impact         | Error code                                | Error No |
|------------|--------------------------|--------------------------|---|----------|
| ERR        | TAI time is out of range | No calculation performed | <code>XL_CFI_TIME_REF_INIT_TAI_ERR</code> | 0        |
| ERR        | UTC time is out of range | No calculation performed | <code>XL_CFI_TIME_REF_INIT_UTC_ERR</code> | 1        |
| ERR        | UT1 time is out of range | No calculation performed | <code>XL_CFI_TIME_REF_INIT_UT1_ERR</code> | 2        |

|     |  |                          |                                  |   |
|-----|--|--------------------------|----------------------------------|---|
|     |  |                          | T1_ERR                           |   |
| ERR | GPS time is out of range                 | No calculation performed | XL_CFI_TIME_REF_INIT_GPS_ERR     | 3 |
| ERR | Absolute orbit number is negative        | No calculation performed | XL_CFI_TIME_REF_INIT_ORBNUM_ERR  | 4 |
| ERR | Elapsed time since ANX is negative       | No calculation performed | XL_CFI_TIME_REF_INIT_ANXTIME_ERR | 5 |
| ERR | Orbit duration is negative               | No calculation performed | XL_CFI_TIME_REF_INIT_ORBDUR_ERR  | 6 |
| ERR | ANX time is bigger than orbit duration   | No calculation performed | XL_CFI_TIME_REF_INIT_COMP_ERR    | 7 |
| ERR | Memory allocation error                  | No calculation performed | XL_CFI_TIME_REF_INIT_MEMORY_ERR  | 8 |
| ERR | Time reference ID is already initialized | No calculation performed | XL_CFI_TIME_REF_INIT_STATUS_ERR  | 9 |

## 7.3 xl\_time\_id\_init

### 7.3.1 Overview

The `xl_time_id_init` CFI function initializes time correlations between TAI, UTC, UT1 and GPS times using any of the following data:

- Set of data read from files (see Table 9 for the allowed file types)
- Data set by the user for the time correlations

### 7.3.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long time_model, time_init_mode, time_ref;
    long orbit0, orbit1;
    double time0, time1, val_time0, val_time1;
    xl_time_id_init_data **init_data;

    status = xl_time_id_init (&time_model,
                             &init_data,
                             &time_init_mode, &time_ref,
                             &time0, &time1,
                             &orbit0, &orbit1,
                             /* output */
                             &val_time0, &val_time1,
                             &time_id,
                             ierr);
}
```

Note that input time vector must be indexed using the existing enumeration for time references.

The `XL_NUM_ERR_TIME_ID_INIT` constant is defined in the file `explorer_lib.h`.

### 7.3.3 Input parameters

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

*Table 17: Input parameters of xl\_time\_id\_init function*

| C name         | C type                 | Array Element | Description (Reference)                         | Unit (Format) | Allowed Range                                      |
|----------------|------------------------|---------------|---|---------------|--|
| time_model     | long *                 | -             | Time model ID                                   | -             | Complete except XL_TIMEMOD_USER or XL_TIMEMOD_AUTO |
| init_data      | xl_time_id_init_data * | -             | Structure with data for the time initialization | -             | -  |
| time_init_mode | long *                 | -             | Flag for selecting the time                     | -             | Select either:                                     |

|          |         |   |   |                                     |  |
|----------|---------|---|---|-------------------------------------|--|
|          |         |   | range of the initialisation: It could be the whole file ( <i>XL_SEL_FILE</i> ), the orbit range given by orbit0-orbit1 ( <i>XO_SEL_ORBIT</i> ) or the time range given by time0-time1( <i>XO_SEL_TIME</i> ) |                                     | <ul style="list-style-type: none"> <li>· <i>XL_SEL_FILE</i></li> <li>· <i>XL_SEL_ORBIT</i></li> <li>· <i>XL_SEL_TIME</i></li> </ul> <ul style="list-style-type: none"> <li>- <i>XL_SEL_ORBIT</i> is not allowed for IERS Bulletins (any format) nor DORIS Navigator files</li> <li>- <i>XL_SEL_ORBIT</i> and <i>XL_SEL_TIME</i> are not enabled for OSF</li> </ul> |
| time_ref | long *  | - | Time reference ID   | -                                   | Complete.<br>If the input file is a DORIS Navigator file and the time_init_mode is <i>XL_SEL_TIME</i> , then only time_ref allowed is <i>XL_TIME_UTC</i> .   |
| time0    | double* | - | If:<br><i>time_init_mode=XL_SEL_TIME</i> Start of the time range defined by [time0,time1]   | Decimal days<br>(Processing format) | [-18262.0,36524.0]   |
| time1    | double* | - | If:<br><i>time_init_mode=XL_SEL_TIME</i> End of the time range defined by [time0,time1]   | Decimal days<br>(Processing format) | [-18262.0,36524.0] > time0   |
| orbit0   | long*   | - | If:<br><i>time_init_mode=XL_SEL_ORBIT</i> Absolute orbit number corresponding to the start of the time range defined by [ <i>ANX<sub>orbit0</sub></i> , <i>ANX<sub>orbit1+1</sub></i> ]                     | -                                   | >= 0   |
| orbit1   | long*   | - | If:<br><i>time_init_mode=XL_SEL_ORBIT</i> Absolute orbit number corresponding to the end of the time range defined by [ <i>ANX<sub>orbit0</sub></i> , <i>ANX<sub>orbit1+1</sub></i> ]                       | -                                   | >orbit0  |

### 7.3.4 Output parameters

The output parameters of the *xl\_time\_id\_init* CFI function are:

**Table 18: Output parameters of *xl\_time\_id\_init* function**

| C name                 | C type  | Array Element | Description (Reference)                                   | Unit (Format)                       | Allowed Range      |
|------------------------|---------|---------------|---|-------------------------------------|--------------------|
| <i>xl_time_id_init</i> | long    | -             | Status flag   | -                                   | -                  |
| <i>val_time0</i>       | double* | -             | Validity start time of the initialization (see table 10). | Decimal days<br>(Processing format) | [-18262.0,36524.0] |



|           |             |   |   |                                  |                    |
|-----------|-------------|---|---|----------------------------------|--------------------|
| val_time1 | double*     | - | Validity end time of the initialization (see table 10). | Decimal days (Processing format) | [-18262.0,36524.0] |
| time_id   | xl_time_id* | - | Structure with the time correlations.                   | -                                | -                  |
| lerr      | long        | - | Error vector  | -                                | -                  |

### 7.3.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_time\_id\_init** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 by translating the error vector returned by the **xl\_time\_id\_init** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 19: Error messages of xl\_time\_id\_init function**

| Error type | Error message                            | Cause and impact         | Error code                         | Error No |
|------------|--|--------------------------|------------------------------------|----------|
| ERR        | Time reference ID is already initialized | No calculation performed | XL_CFI_TIME_ID_INIT_STATUS_ERR     | 0        |
| ERR        | Time model ID is not correct             | No calculation performed | XL_CFI_TIME_ID_INIT_TIME_MODEL_ERR | 1        |
| ERR        | Time init mode ID is not correct         | No calculation performed | XL_CFI_TIME_ID_INIT_INIT_MODE_ERR  | 2        |
| ERR        | Time reference ID is not correct         | No calculation performed | XL_CFI_TIME_ID_INIT_TIME_ERR       | 3        |
| ERR        | Reference start time out of limits       | No calculation performed | XL_CFI_TIME_ID_INIT_DAY_0_ERR      | 4        |
| ERR        | Reference end time out of limits         | No calculation performed | XL_CFI_TIME_ID_INIT_DAY_1_ERR      | 5        |
| ERR        | Wrong reference time range               | No calculation performed | XL_CFI_TIME_ID_INIT_DAY_RANGE_ERR  | 6        |
| ERR        | Reference start orbit is negative        | No calculation performed | XL_CFI_TIME_ID_INIT_ORB_0_ERR      | 7        |
| ERR        | Reference end orbit is negative          | No calculation performed | XL_CFI_TIME_ID_INIT_ORB_1_ERR      | 8        |
| ERR        | Wrong reference orbit range              | No calculation performed | XL_CFI_TIME_ID_INIT_ORB_RANGE_ERR  | 9        |
| ERR        | No data in the input structures          | No calculation performed | XL_CFI_TIME_ID_INIT_NO_DATA_ERR    | 10       |

|      |   |  |   |    |
|------|---|--|---|----|
| ERR  | input data structure contains data for diferent file types                        | No calculation performed   | XL_CFI_TIME_ID_INIT_INCONSISTENT_FILES_ERR      | 11 |
| ERR  | Memory allocation error   | No calculation performed   | XL_CFI_TIME_ID_INIT_MEMORY_ERR                  | 12 |
| ERR  | No data in the input structure for the requested initialization range             | No calculation performed   | XL_CFI_TIME_ID_INIT_NO_DATA_IN_RANGE_ERR        | 13 |
| ERR  | Error trying to initialize the time id  | No calculation performed   | XL_vTIME_ID_INIT_TIME_CORR_INIT_ERR             | 14 |
| ERR  | Error merging the input set of files  | No calculation performed   | XL_CFI_TIME_ID_INIT_LOAD_TIME_INIT_LIST_ERR,ERR | 15 |
| ERR  | Input Time model is inconsistent with the data file type                          | No calculation performed   | XL_CFI_TIME_ID_INIT_WRONG_TIME_MODEL_ERR        | 16 |
| ERR  | Incorrect input data type. It should be XL_FILE_DATA or XL_TIME_CORRELATIONS_DATA | No calculation performed   | XL_CFI_TIME_ID_INIT_WRONG_DATA_TYPE_ERR         | 17 |
| WARN | Time init mode option not enabled for file  | Calculation performed<br>All data in OSF range is used for computation   | XL_CFI_TIME_ID_INIT_INIT_MODE_WARN              | 18 |
| WARN | Only one OSF file is admitted for this initialisation mode                        | Calculation performed with the first OSF data structure in the init_data | XL_CFI_TIME_ID_INIT_ONLY_FIRST_OSF_WARN         | 19 |

## 7.4 xl\_time\_close

### 7.4.1 Overview

The `xl_time_close` CFI function cleans up any memory allocation performed by the initialization functions. A complete calling sequence of the time reference computations is presented in section 4.2..

### 7.4.2 Calling interface

The calling interface of the `xl_time_close` CFI function is the following:

```
#include <explorer_lib.h>
{
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_TIME_CLOSE], status;
    status = xl_time_close (&time_id, ierr);
}
```

### 7.4.3 Input parameters

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

*Table 20: Input parameters of xl\_time\_close function*

| C name  | C type      | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|---------|-------------|---------------|--|---------------|---------------|
| time_id | xl_time_id* | -             | Structure that contains the time correlations. | -             | -             |

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

- Satellite ID: `sat_id`. See [GEN\_SUM].

### 7.4.4 Output parameters

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

*Table 21: Output parameters of xl\_time\_close function*

| C name        | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|---------------|--------|---------------|-------------------------|---------------|---------------|
| xl_time_close | long   | -             | Status flag             | -             | -             |
| ier           | long   | -             | Error vector            | -             | -             |

### 7.4.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_time\_close** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 extended status flag returned by the **xl\_time\_close** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM])

**Table 22: Error messages of xl\_time\_close function**

| Error type | Error message   | Cause and impact         | Error code                     | Error No |
|------------|---|--------------------------|--------------------------------|----------|
| ERR        | The Time Id is not initialized or it could be in use by another Id. | No calculation performed | XL_CFI_TIME_CLOSE_WRONG_ID_ERR | 0        |

## 7.5 xl\_time\_get\_id\_data

### 7.5.1 Overview

The `xl_time_get_id_data` CFI function returns a data structure containing the data used for the time initialisation.

### 7.5.2 Calling interface

The calling interface of the `xl_time_get_id_data` CFI function is the following:

```
#include <explorer_lib.h>
{
    xl_time_id time_id;
    xl_time_id_data data;
    long status;
    status = xl_time_get_id_data (&time_id, &data);
}
```

### 7.5.3 Input parameters

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

*Table 23: Input parameters of xl\_time\_get\_id\_data function*

| C name  | C type      | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|---------|-------------|---------------|--|---------------|---------------|
| time_id | xl_time_id* | -             | Structure that contains the time correlations. | -             | -             |

### 7.5.4 Output parameters

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

*Table 24: Output parameters of xl\_time\_get\_id\_data function*

| C name              | C type          | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|---------------------|-----------------|---------------|-------------------------|---------------|---------------|
| xl_time_get_id_data | long            | -             | Status flag             | -             | -             |
| data                | xl_time_id_data | -             | Time ID data            | -             | -             |

The data structure `xl_time_id_get_id_data` can be seen in Table 8.

### **7.5.5 Warnings and errors**

This function does not return any error/warning code. Only the status of the function indicates if the execution was correct or not.

The possible causes of error are:

- The time\_id was not initialised.

## 7.6 xl\_time\_set\_id\_data

### 7.6.1 Overview

The `xl_time_set_id_data` CFI function changes the time correlations that are stored within a `time_id`.

### 7.6.2 Calling interface

The calling interface of the `xl_time_set_id_data` CFI function is the following:

```
#include <explorer_lib.h>
{
    xl_time_id time_id;
    xl_time_id_data data;
    long status;
    status = xl_time_set_time_id (&time_id, &data);
}
```

### 7.6.3 Input parameters

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

*Table 25: Input parameters of xl\_time\_set\_id\_data function*

| C name  | C type      | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|---------|-------------|---------------|--|---------------|---------------|
| time_id | xl_time_id* | -             | Structure that contains the time correlations (input/output parameter) | -             | -             |

### 7.6.4 Output parameters

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

*Table 26: Output parameters of xl\_time\_set\_id\_data function*

| C name              | C type      | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|---------------------|-------------|---------------|--|---------------|---------------|
| xl_time_set_id_data | long        | -             | Status flag  | -             | -             |
| time_id             | xl_time_id* | -             | Structure that contains the time correlations (input/output parameter) | -             | -             |

---

|      |                 |   |              |   |   |
|------|-----------------|---|--------------|---|---|
|      |                 |   | parameter)   |   |   |
| data | xl_time_id_data | - | Time ID data | - | - |

The data structure **xl\_time\_set\_id\_data** can be seen in Table 8.

### **7.6.5 Warnings and errors**

This function does not return any error/warning code. Only the status of the function indicates if the execution was correct or not.

The possible causes of error are:

- The time\_id was not initialised.



## 7.7 xl\_run\_init

### 7.7.1 Overview

The `xl_run_init` CFI function groups into a single *id* the *satellite Id*, the *time Id* and the *model\_id*, creating a *run Id*.

### 7.7.2 Calling interface

The calling interface of the `xl_run_init` CFI function is the following:

```
#include <explorer_lib.h>
{
    long sat_id, run_id;
    xl_model_id model_id = {NULL};
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_RUN_INIT], status;
    status = xl_run_init (&sat_id, &time_id, &model_id,
                        &run_id, ierr);
}
```

### 7.7.3 Input parameters

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

**Table 27: Input parameters of `xl_run_init` function**

| C name   | C type       | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|----------|--------------|---------------|--|---------------|---------------|
| sat_id   | long *       | -             | Satellite ID                                   | -             | Complete      |
| time_id  | xl_time_id*  | -             | Structure that contains the time correlations. | -             | -             |
| model_id | xl_model_id* | -             | Model ID                                       | -             | -             |

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

- Satellite ID: `sat_id`. See [GEN\_SUM].

### 7.7.4 Output parameters

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

**Table 28: Output parameters of xl\_run\_init function**

| C name      | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|-------------|--------|---------------|-------------------------|---------------|---------------|
| xl_run_init | long   | -             | Status flag             | -             | -             |
| run_id      | long * | -             | Run ID                  | -             | >=0           |
| ierr        | long   | -             | Error vector            | -             | -             |

### 7.7.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_run\_init** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 extended status flag returned by the **xl\_run\_init** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM])

**Table 29: Error messages of xl\_run\_init function**

| Error type | Error message                               | Cause and impact         | Error code                        | Error No |
|------------|---|--------------------------|-----------------------------------|----------|
| ERR        | Maximum number of initializations reached   | No calculation performed | XL_CFI_RUN_INIT_MAX_INIT_ERR      | 0        |
| ERR        | Satellite ID is not correct                 | No calculation performed | XL_CFI_RUN_INIT_SAT_ERR           | 1        |
| ERR        | Time ID is not initialized                  | No calculation performed | XL_CFI_RUN_INIT_TIME_INIT_ERR     | 2        |
| ERR        | Memory allocation error                     | No calculation performed | XL_CFI_RUN_INIT_MEMORY_ERR        | 3        |
| ERR        | Inconsistency between Ids within the run_id | No calculation performed | XL_CFI_RUN_INIT_INCONSISTENCY_ERR | 4        |
| ERR        | Could not lock other execution threads      | No calculation performed | XL_CFI_RUN_INIT_LOCK_ERR          | 5        |
| ERR        | Could not unlock other execution threads    | No calculation performed | XL_CFI_RUN_INIT_UNLOCK_ERR        | 6        |

## 7.8 xl\_run\_get\_ids

### 7.8.1 Overview

The `xl_run_get_ids` CFI function returns the *ids* being used.

### 7.8.2 Calling interface

The calling interface of the `xl_run_get_ids` CFI function is the following:

```
#include <explorer_lib.h>
{
    long sat_id, run_id;
    xl_time_id time_id = {NULL};
    xl_model_id model_id = {NULL};
    xl_run_get_ids (&run_id,
                  &sat_id, &time_id &model_id);
}
```

### 7.8.3 Input parameters

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

*Table 30: Input parameters of xl\_run\_get\_ids function*

| C name | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------|--------|---------------|-------------------------|---------------|---------------|
| run_id | long * | -             | Run ID                  | -             | >=0           |

### 7.8.4 Output parameters

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

*Table 31: Output parameters of xl\_run\_get\_ids function*

| C name         | C type       | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|----------------|--------------|---------------|--|---------------|---------------|
| xl_run_get_ids | void         | -             | -  | -             | -             |
| sat_id         | long*        | -             | Satellite ID                                   | -             | -             |
| time_id        | xl_time_id*  | -             | Structure that contains the time correlations. | -             | -             |
| model_id       | xl_model_id* | -             | Model ID                                       | -             | -             |

### **7.8.5 Warnings and errors**

Next table lists the possible error messages that can be returned by the **xl\_run\_get\_ids** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 extended status flag returned by the **xl\_run\_get\_ids** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM])

TBW

## 7.9 xl\_run\_close

### 7.9.1 Overview

The `xl_run_close` CFI function cleans up any memory allocation performed by the initialization functions.

### 7.9.2 Calling interface

The calling interface of the `xl_run_close` CFI function is the following:

```
#include <explorer_lib.h>
{
    long run_id;
    xl_run_close (&run_id);
}
```

### 7.9.3 Input parameters

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

*Table 32: Input parameters of xl\_run\_close function*

| C name | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------|--------|---------------|-------------------------|---------------|---------------|
| run_id | long * | -             | Run ID                  | -             | >=0           |

### 7.9.4 Output parameters

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

*Table 33: Output parameters of xl\_run\_close function*

| C name       | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------------|--------|---------------|-------------------------|---------------|---------------|
| xl_run_close | void   | -             | -                       | -             | -             |

### 7.9.5 Warnings and errors

No errors have been envisaged for `xl_run_close`.

## 7.10xl\_time\_ascii\_to\_ascii

### 7.10.1Overview

The `xl_time_ascii_to_ascii` CFI function transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a different ASCII format and/or reference (TAI, UTC, UT1 or GPS).

### 7.10.2Calling Interface

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

```
#include <explorer_lib.h>
{
    long ascii_id_in, ascii_id_out;
    long time_ref_in, time_ref_out;
    char ascii_in[XL_TIME_ASCII_DIM_MAX];
    char ascii_out[XL_TIME_ASCII_DIM_MAX];
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_ASCII_ASCII], status;

    status = xl_time_ascii_to_ascii(&time_id, &ascii_id_in,
                                   &time_ref_in, ascii_in, &ascii_id_out,
                                   &time_ref_out, ascii_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_ascii_to_ascii_run(&run_id, &ascii_id_in,
                                       &time_ref_in, ascii_in, &ascii_id_out,
                                       &time_ref_out, ascii_out, ierr);
}
```

The `XL_TIME_ASCII_DIM_MAX` and `XL_NUM_ERR_ASCII_ASCII` constants are defined in the file `explorer_lib.h`.

### 7.10.3 Input Parameters

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

**Table 34: Input parameters of `xl_time_ascii_to_ascii` function**

| C name                    | C type                   | Array Element           | Description (Reference)                        | Unit (Format)           | Allowed Range                         |
|---------------------------|--------------------------|-------------------------|--|-------------------------|---------------------------------------|
| <code>time_id</code>      | <code>xl_time_id*</code> | -                       | Structure that contains the time correlations. | -                       | -                                     |
| <code>ascii_id_in</code>  | <code>long *</code>      | -                       | ASCII format ID                                | -                       | Complete                              |
| <code>time_ref_in</code>  | <code>long *</code>      | -                       | Time reference ID                              | -                       | Complete                              |
| <code>ascii_in</code>     | <code>char</code>        | See Table 4 and Table 5 | Time in ASCII format                           | See Table 4 and Table 5 | See Table 4 and Table 5               |
| <code>ascii_id_out</code> | <code>long *</code>      | -                       | ASCII format ID                                | -                       | Complete                              |
| <code>time_ref_out</code> | <code>long *</code>      | -                       | Time reference ID                              | -                       | Any except <code>XL_TIME_UNDEF</code> |

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

- ASCII format ID: `ascii_id_in` and `ascii_id_out`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].

It is important to point out the usage of the `time_ref_in` parameter in the frame of the current function:

- If `time_ref_in` input parameter is defined, it shall be used by the function.
- If `time_ref_in` input parameter is undefined, it shall be used the time reference part from the `ascii` format string. In case this is omitted, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.10.4 Output Parameters

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

**Table 35: Output parameters of `xl_time_ascii_to_ascii`**

| C name                              | C type            | Array Element           | Description (Reference) | Unit (Format)           | Allowed Range           |
|-------------------------------------|-------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| <code>xl_time_ascii_to_ascii</code> | <code>long</code> | -                       | Status flag             | -                       | -                       |
| <code>ascii_out</code>              | <code>char</code> | See Table 4 and Table 5 | Time in ASCII format    | See Table 4 and Table 5 | See Table 4 and Table 5 |
| <code>ierr</code>                   | <code>long</code> | -                       | Error vector            | -                       | -                       |

### 7.10.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_ascii_to_ascii` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_ascii_to_ascii` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM])

**Table 36: Error messages of `xl_time_ascii_to_ascii` function**

| Error type | Error message   | Cause and impact         | Error code                                    | Error No |
|------------|---|--------------------------|---|----------|
| ERR        | Input ascii format ID is not correct                                  | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_IN_ERR          | 0        |
| ERR        | Input time reference ID is not correct                                | No calculation performed | XL_CFI_TIME_ASCII_ASCII_TIME_IN_ERR           | 1        |
| ERR        | Input ascii format is not correct                                     | No calculation performed | XL_CFI_TIME_ASCII_ASCII_FORMAT_IN_ERR         | 2        |
| ERR        | Input time reference inconsistent with the time reference in the date | No calculation performed | XL_CFI_TIME_ASCII_ASCII_REF_INC_IN_ERR        | 3        |
| ERR        | Output ascii format ID is not correct                                 | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_OUT_ERR         | 4        |
| ERR        | Output time reference ID is not correct                               | No calculation performed | XL_CFI_TIME_ASCII_ASCII_TIME_OUT_ERR          | 5        |
| ERR        | Input ascii year is out of range                                      | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_YEAR_IN_ERR     | 6        |
| ERR        | Input ascii month is out of range                                     | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_MONTH_IN_ERR    | 7        |
| ERR        | Input ascii day is out of range                                       | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_DAY_IN_ERR      | 8        |
| ERR        | Input ascii hour is out of range                                      | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_HOUR_IN_ERR     | 9        |
| ERR        | Input ascii minutes are out of range                                  | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_MIN_IN_ERR      | 10       |
| ERR        | Input ascii seconds are out of range                                  | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_SEC_IN_ERR      | 11       |
| ERR        | Input ascii microseconds are out of range                             | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_MICROSEC_IN_ERR | 12       |
| ERR        | Internal error: Input Gregorian date to MJD transformation failed     | No calculation performed | XL_CFI_TIME_ASCII_ASCII_MJD_IN_ERR            | 13       |
| ERR        | Internal error: Output ascii MJD is out of range                      | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_MJD_OUT_ERR     | 14       |
| ERR        | Internal error: Output ascii year is out of range                     | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_YEAR_OUT_ERR    | 15       |
| ERR        | Internal error: Output ascii month is out of range                    | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_MONTH_OUT_ERR   | 16       |
| ERR        | Internal error: Output ascii day is out of range                      | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_DAY_OUT_ERR     | 17       |
| ERR        | Internal error: Output ascii hour is out of range                     | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_HOUR_OUT_ERR    | 18       |
| ERR        | Internal error: Output ascii minutes are out of range                 | No calculation performed | XL_CFI_TIME_ASCII_ASCII_ASCII_MIN_OUT_ERR     | 19       |



|      |  |  |  |    |
|------|--|--|--|----|
| ERR  | Internal error: Output ascii seconds are out of range  | No calculation performed                           | XL_CFI_TIME_ASCII_ASCII_ASCII_SEC_OUT_ERR        | 20 |
| ERR  | Internal error: Output ascii microseconds are out of range   | No calculation performed                           | XL_CFI_TIME_ASCII_ASCII_ASCII_MICROSEC_OUT_ERR   | 21 |
| ERR  | Internal error: Output ascii format is not correct   | No calculation performed                           | XL_CFI_TIME_ASCII_ASCII_ASCII_FORMAT_OUT_ERR     | 22 |
| ERR  | Time reference not initialised   | No calculation performed                           | XL_CFI_TIME_ASCII_ASCII_ASCII_REF_INIT_ERR       | 23 |
| WARN | Time out of initialization range   | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_ASCII_ASCII_REF_INIT_WARN      | 24 |
| WARN | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_ASCII_ASCII_BUL_A_FORMULA_WARN | 25 |
| WARN | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_ASCII_ASCII_BUL_B_A_GAP_WARN   | 26 |
| WARN | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_ASCII_ASCII_VALIDITY_WARN      | 27 |

## 7.11 xl\_time\_ascii\_to\_processing

### 7.11.1 Overview

The `xl_time_ascii_to_processing` CFI function transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.11.2 Calling Interface

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

```
#include <explorer_lib.h>
{
    long ascii_id_in, proc_id_out;
    long time_ref_in, time_ref_out;
    char ascii_in[XL_TIME_ASCII_DIM_MAX];
    double processing_out;
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_ASCII_PROC], status;

    status = xl_time_ascii_to_processing(&time_id, &ascii_id_in,
                                       &time_ref_in, ascii_in, &proc_id_out,
                                       &time_ref_out, &processing_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_ascii_to_processing_run(&run_id, &ascii_id_in,
                                           &time_ref_in, ascii_in, &proc_id_out,
                                           &time_ref_out, &processing_out, ierr);
}
```

The `XL_TIME_ASCII_DIM_MAX` and `XL_NUM_ERR_ASCII_PROC` constants are defined in the file `explorer_lib.h`.

### 7.11.3 Input Parameters

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

*Table 37: Input parameters of `xl_time_ascii_to_processing` function*

| C name                    | C type                   | Array Element           | Description (Reference)                        | Unit (Format)           | Allowed Range            |
|---------------------------|--------------------------|-------------------------|--|-------------------------|--------------------------|
| <code>time_id</code>      | <code>xl_time_id*</code> | -                       | Structure that contains the time correlations. | -                       | -                        |
| <code>ascii_id_in</code>  | <code>long *</code>      | -                       | ASCII format ID                                | -                       | Complete                 |
| <code>time_ref_in</code>  | <code>long *</code>      | -                       | Time reference ID                              | -                       | Complete                 |
| <code>ascii_in</code>     | <code>char</code>        | See Table 4 and Table 5 | Time in ASCII format                           | See Table 4 and Table 5 | See Table 4 and Table 5  |
| <code>proc_id_out</code>  | <code>long *</code>      | -                       | Processing format ID                           | -                       | Complete                 |
| <code>time_ref_out</code> | <code>long *</code>      | -                       | Time reference ID                              | -                       | Any except XL_TIME_UNDEF |

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

- ASCII format ID: `ascii_id_in`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].
- Processing format ID: `proc_id_out`. Current document, section 6.2

It is important to point out the usage of the `time_ref_in` parameter in the frame of the current function:

- If `time_ref_in` input parameter is defined, it shall be used by the function.
- If `time_ref_in` input parameter is undefined, it shall be used the time reference part from the `ascii` format string. In case this is omitted, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.11.4 Output Parameters

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

*Table 38: Output parameters of `xl_time_ascii_to_processing`*

| C name                                   | C type               | Array Element | Description (Reference)   | Unit (Format)                      | Allowed Range      |
|--|----------------------|---------------|---------------------------|------------------------------------|--------------------|
| <code>xl_time_ascii_to_processing</code> | <code>long</code>    | -             | Status flag               | -                                  | -                  |
| <code>processing_out</code>              | <code>double*</code> | -             | Time in Processing Format | Decimal days, MJD2000 (Processing) | [-18262.0,36524.0] |
| <code>ierr</code>                        | <code>long</code>    | -             | Error vector              | -                                  | -                  |

### 7.11.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_ascii_to_processing` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_ascii_to_processing` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 39: Error messages of `xl_time_ascii_to_processing` function**

| Error type | Error message   | Cause and impact                                      | Error code                                    | Error No |
|------------|---|---|---|----------|
| ERR        | Input ascii format ID is not correct  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_ASCII_IN_ERR       | 0        |
| ERR        | Input time reference ID is not correct  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_TIME_IN_ERR        | 1        |
| ERR        | Input format is not correct   | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_FORMAT_IN_ERR      | 2        |
| ERR        | Input time reference inconsistent with the time reference in the date                           | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_REF_INC_IN_ERR     |          |
| ERR        | Output processing format ID is not correct  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_PROC_OUT_ERR       | 4        |
| ERR        | Output time reference ID is not correct   | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_TIME_OUT_ERR       | 5        |
| ERR        | Year is out of range  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_YEAR_ERR           | 6        |
| ERR        | Month is out of range   | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_MONTH_ERR          | 7        |
| ERR        | Day is out of range   | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_DAY_ERR            | 8        |
| ERR        | Hour is out of range  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_HOUR_ERR           | 9        |
| ERR        | Minutes are out of range  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_MIN_ERR            | 10       |
| ERR        | Seconds are out of range  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_SEC_ERR            | 11       |
| ERR        | Microseconds are out of range   | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_MICROSEC_ERR       | 12       |
| ERR        | Internal Error: Input Gregorian date to MJD transformation failed                               | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_MJD_ERR            | 13       |
| ERR        | Time reference not initialised  | No calculation performed                              | XL_CFI_TIME_ASCII_PRO<br>C_REF_INIT_ERR       | 14       |
| WARN       | Time out of initialization range  | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_ASCII_PRO<br>C_REF_INIT_WARN      | 15       |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_ASCII_PRO<br>C_BUL_A_FORMULA_WARN | 16       |

|      |  |  |   |    |
|------|--|--|---|----|
| WARN | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_PROC_BUL_B_A_GAP_WARN | 17 |
| WARN | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_PROC_VALIDITY_WARN    | 18 |

## 7.12xl\_time\_ascii\_to\_transport

### 7.12.1Overview

The `xl_time_ascii_to_transport` CFI function transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

### 7.12.2Calling Interface

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

```
#include <explorer_lib.h>
{
    long ascii_id_in, trans_id_out;
    long time_ref_in, time_ref_out;
    char ascii_in[XL_TIME_ASCII_DIM_MAX];
    long transport_out[XL_TIME_TRANS_DIM_MAX];
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_ASCII_TRANS], status;

    status = xl_time_ascii_to_transport(&time_id, &ascii_id_in,
                                       &time_ref_in, ascii_in, &trans_id_out,
                                       &time_ref_out, transport_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_ascii_to_transport_run(&run_id, &ascii_id_in,
                                           &time_ref_in, ascii_in, &trans_id_out,
                                           &time_ref_out, transport_out, ierr);
}
```

The `XL_TIME_TRANS_DIM_MAX`, `XL_TIME_ASCII_DIM_MAX`, `XL_NUM_ERR_ASCII_TRANS` constants are defined in the file `explorer_lib.h`.

### 7.12.3 Input Parameters

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

**Table 40: Input parameters of `xl_time_ascii_to_transport` function**

| C name                    | C type                   | Array Element           | Description (Reference)                        | Unit (Format)           | Allowed Range            |
|---------------------------|--------------------------|-------------------------|--|-------------------------|--------------------------|
| <code>time_id</code>      | <code>xl_time_id*</code> | -                       | Structure that contains the time correlations. | -                       | -                        |
| <code>ascii_id_in</code>  | <code>long *</code>      | -                       | ASCII format ID                                | -                       | Complete                 |
| <code>time_ref_in</code>  | <code>long *</code>      | -                       | Time reference ID                              | -                       | Complete                 |
| <code>ascii_in</code>     | <code>char</code>        | See Table 4 and Table 5 | Time in ASCII format                           | See Table 4 and Table 5 | See Table 4 and Table 5  |
| <code>trans_id_out</code> | <code>long *</code>      | -                       | Transport format ID                            | -                       | Complete                 |
| <code>time_ref_out</code> | <code>long *</code>      | -                       | Time reference ID                              | -                       | Any except XL_TIME_UNDEF |

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

- Satellite ID: `sat_id`. See [GEN\_SUM].
- ASCII format ID: `trans_id_in`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].
- Transport format ID: `trans_id_out`. Current document, section 6.2.

It is important to point out the usage of the `time_ref_in` parameter in the frame of the current function:

- If `time_ref_in` input parameter is defined, it shall be used by the function.
- If `time_ref_in` input parameter is undefined, it shall be used the time reference part from the `ascii` format string. In case this is omitted, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.12.4 Output Parameters

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

**Table 41: Output parameters of `xl_time_ascii_to_transport`**

| C name                                  | C type            | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|---|-------------------|---------------|--------------------------|---------------|---------------|
| <code>xl_time_ascii_to_transport</code> | <code>long</code> | -             | Status flag              | -             | -             |
| <code>transport_out[dim]</code>         | <code>long</code> | See Table 3   | Time in Transport format | See Table 3   | See Table 3   |
| <code>ierr</code>                       | <code>long</code> | -             | Error vector             | -             | -             |

### 7.12.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_ascii_to_transport` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_ascii_to_transport` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 42: Error messages of `xl_time_ascii_to_transport` function**

| Error type | Error message   | Cause and impact                                   | Error code                                 | Error No |
|------------|---|--|--|----------|
| ERR        | Input ascii format ID is not correct  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_ASCII_IN_ERR       | 0        |
| ERR        | Input time reference ID is not correct  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_TIME_IN_ERR        | 1        |
| ERR        | Input format is not correct   | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_FORMAT_IN_ERR      | 2        |
| ERR        | Input time reference inconsistent with the time reference in the date                           | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_REF_INC_IN_ERR     | 3        |
| ERR        | Output transport format ID is not correct   | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_TRANS_OUT_ERR      | 4        |
| ERR        | Output time reference ID is not correct   | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_TIME_OUT_ERR5      | 5        |
| ERR        | Year is out of range  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_YEAR_ERR           | 6        |
| ERR        | Month is out of range   | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_MONTH_ERR          | 7        |
| ERR        | Day is out of range   | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_DAY_ERR            | 8        |
| ERR        | Hour is out of range  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_HOUR_ERR           | 9        |
| ERR        | Minutes are out of range  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_MIN_ERR            | 10       |
| ERR        | Seconds are out of range  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_SEC_ERR            | 11       |
| ERR        | Microseconds are out of range   | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_MICROSEC_ERR       | 12       |
| ERR        | Internal Error: Input Gregorian date to MJD transformation failed                               | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_MJD_ERR            | 13       |
| ERR        | Time reference not initialised  | No calculation performed                           | XL_CFI_TIME_ASCII_TRANS_REF_INIT_ERR       | 14       |
| WARN       | Time out of initialization range  | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_TRANS_REF_INIT_WARN      | 15       |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_TRANS_BUL_A_FORMULA_WARN | 16       |



|      |  |  |  |    |
|------|--|--|--|----|
| WARN | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_TRAN<br>S_BUL_B_A_GAP_WARN | 17 |
| WARN | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_ASCII_TRAN<br>S_VALIDITY_WARN    | 18 |

## 7.13 xl\_time\_processing\_to\_ascii

### 7.13.1 Overview

The `xl_time_processing_to_ascii` CFI function transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.13.2 Calling Interface

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

```
#include <explorer_lib.h>
{
    long proc_id_in, ascii_id_out;
    long time_ref_in, time_ref_out;
    double processing_in;
    char ascii_out[XL_TIME_ASCII_DIM_MAX];
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_PROC_ASCII], status;

    status = xl_time_processing_to_ascii(&time_id, &proc_id_in,
                                        &time_ref_in, &processing_in, &ascii_id_out,
                                        &time_ref_out, ascii_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_processing_to_ascii_run(&run_id, &proc_id_in,
                                            &time_ref_in, &processing_in, &ascii_id_out,
                                            &time_ref_out, ascii_out, ierr);
}
```

### 7.13.3 Input Parameters

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

**Table 43: Input parameters of `xl_time_processing_to_ascii` function**

| C name                     | C type                   | Array Element | Description (Reference)                        | Unit (Format)                      | Allowed Range                         |
|----------------------------|--------------------------|---------------|--|------------------------------------|---------------------------------------|
| <code>time_id</code>       | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -                                  | -                                     |
| <code>proc_id_in</code>    | <code>long *</code>      | -             | Processing format ID                           | -                                  | Complete                              |
| <code>time_ref_in</code>   | <code>long *</code>      | -             | Time reference ID                              | -                                  | Any except <code>XL_TIME_UNDEF</code> |
| <code>processing_in</code> | <code>double*</code>     | -             | Time in Processing Format                      | Decimal days, MJD2000 (Processing) | [-18262.0,36524.0]                    |
| <code>ascii_id_out</code>  | <code>long *</code>      | -             | ASCII format ID                                | -                                  | Complete                              |
| <code>time_ref_out</code>  | <code>long *</code>      | -             | Time reference ID                              | -                                  | Any except <code>XL_TIME_UNDEF</code> |

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

- Processing format ID: `proc_id_in`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].
- ASCII format ID: `ascii_id_out`. Current document, section 6.2.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.13.4 Output Parameters

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

**Table 44: Output parameters of `xl_time_processing_to_ascii`**

| C name                                   | C type            | Array Element           | Description (Reference) | Unit (Format)           | Allowed Range           |
|--|-------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| <code>xl_time_processing_to_ascii</code> | <code>long</code> | -                       | Status flag             | -                       | -                       |
| <code>ascii_out</code>                   | <code>char</code> | See Table 4 and Table 5 | Time in ASCII format    | See Table 4 and Table 5 | See Table 4 and Table 5 |
| <code>ierr</code>                        | <code>long</code> | -                       | Error vector            | -                       | -                       |

### 7.13.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_processing_to_ascii` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_processing_to_ascii` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM])

**Table 45: Error messages of `xl_time_processing_to_ascii` function**

| Error type | Error message  | Cause and impact                                   | Error code                                | Error No |
|------------|--|--|---|----------|
| ERR        | Input processing format ID is not correct  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_PROC_IN_ERR        | 0        |
| ERR        | Input time reference ID is not correct   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_TIME_IN_ERR        | 1        |
| ERR        | Input days out of range  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_DAY_ERR            | 2        |
| ERR        | Output ascii format ID is not correct  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_OUT_ERR      | 3        |
| ERR        | Output time reference ID is not correct  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_TIME_OUT_ERR       | 4        |
| ERR        | Internal error: Output ascii MJD is out of range   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_MJD_ERR      | 5        |
| ERR        | Internal error: Output ascii year is out of range  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_YEAR_ERR     | 6        |
| ERR        | Internal error: Output ascii month is out of range   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_MONTH_ERR    | 7        |
| ERR        | Internal error: Output ascii day is out of range   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_DAY_ERR      | 8        |
| ERR        | Internal error: Output ascii hour is out of range  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_HOUR_ERR     | 9        |
| ERR        | Internal error: Output ascii minutes are out of range  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_MIN_ERR      | 10       |
| ERR        | Internal error: Output ascii seconds are out of range  | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_SEC_ERR      | 11       |
| ERR        | Internal error: Output ascii microseconds are out of range   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_ASCII_MICROSEC_ERR | 12       |
| ERR        | Internal error: Output ascii format is not correct   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_FORMAT_OUT_ERR     | 13       |
| ERR        | Time reference not initialised   | No calculation performed                           | XL_CFI_TIME_PROC_ASCII_REF_INIT_ERR       | 14       |
| WARN       | Time out of initialization range   | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_ASCII_REF_INIT_WARN      | 15       |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_ASCII_BUL_A_FORMULA_WARN | 16       |
| WARN       | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_ASCII_BUL_B_A_GAP_WARN   | 17       |
| WARN       | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_ASCII_VALIDITY_WARN      | 18       |



Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 93

---

## 7.14 **xl\_time\_processing\_to\_processing**

### 7.14.1 **Overview**

The **xl\_time\_processing\_to\_processing** CFI function transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in Processing format with a different reference (TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.14.2 **Calling Interface**

The calling interface of the **xl\_time\_processing\_to\_processing** CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long proc_id_in, proc_id_out;
    long time_ref_in, time_ref_out;
    double processing_in, processing_out;
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_PROC_PROC], status;

    status = xl_time_processing_to_processing(&time_id,
    &proc_id_in,
    &time_ref_in, &processing_in, &proc_id_out,
    &time_ref_out, &processing_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_processing_to_processing_run(&run_id, &proc_id_in,
    &time_ref_in, &processing_in, &proc_id_out,
    &time_ref_out, &processing_out, ierr);
}
```

The `XL_NUM_ERR_PROC_PROC` constant is defined in the file *explorer\_lib.h*.

### 7.14.3 Input Parameters

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

*Table 46: Input parameters of `xl_time_processing_to_processing` function*

| C name                     | C type                   | Array Element | Description (Reference)                        | Unit (Format)                      | Allowed Range                         |
|----------------------------|--------------------------|---------------|--|------------------------------------|---------------------------------------|
| <code>time_id</code>       | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -                                  | -                                     |
| <code>proc_id_in</code>    | <code>long *</code>      | -             | Processing format ID                           | -                                  | Complete                              |
| <code>time_ref_in</code>   | <code>long *</code>      | -             | Time reference ID                              | -                                  | Any except <code>XL_TIME_UNDEF</code> |
| <code>processing_in</code> | <code>double*</code>     | -             | Time in Processing Format                      | Decimal days, MJD2000 (Processing) | [-18262.0,36524.0]                    |
| <code>proc_id_out</code>   | <code>long *</code>      | -             | Processing format ID                           | -                                  | Complete                              |
| <code>time_ref_out</code>  | <code>long *</code>      | -             | Time reference ID                              | -                                  | Any except <code>XL_TIME_UNDEF</code> |

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

- Processing format ID: `proc_id_in` and `proc_id_out`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.14.4 Output Parameters

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

*Table 47: Output parameters of `xl_time_processing_to_processing`*

| C name  | C type               | Array Element | Description (Reference)   | Unit (Format)                      | Allowed Range      |
|---|----------------------|---------------|---------------------------|------------------------------------|--------------------|
| <code>xl_time_processing_to_processing</code> | <code>long</code>    | -             | Status flag               | -                                  | -                  |
| <code>processing_out</code>                   | <code>double*</code> | -             | Time in Processing Format | Decimal days, MJD2000 (Processing) | [-18262.0,36524.0] |
| <code>ierr</code>                             | <code>long</code>    | -             | Error vector              | -                                  | -                  |

### 7.14.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_processing_to_processing` CFI function after translating the returned error vector into the equivalent list of error messages by calling

the function of the EO\_LIB software library **xl\_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 by translating the error vector returned by the **xl\_time\_processing\_to\_processing** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM])

**Table 48: Error messages of xl\_time\_processing\_to\_processing function**

| Error type | Error message  | Cause and impact                                      | Error code                                   | Error No |
|------------|--|---|--|----------|
| ERR        | Input processing format ID is not correct  | No calculation performed                              | XL_CFI_TIME_PROC_PRO<br>C_PROC_IN_ERR        | 0        |
| ERR        | Input time reference ID is not correct   | No calculation performed                              | XL_CFI_TIME_PROC_PRO<br>C_TIME_IN_ERR        | 1        |
| ERR        | Output processing format ID is not correct   | No calculation performed                              | XL_CFI_TIME_PROC_PRO<br>C_PROC_OUT_ERR       | 2        |
| ERR        | Output time reference ID is not correct  | No calculation performed                              | XL_CFI_TIME_PROC_PRO<br>C_TIME_OUT_ERR       | 3        |
| ERR        | Number of days out of range  | No calculation performed                              | XL_CFI_TIME_PROC_PRO<br>C_DAY_ERR            | 4        |
| ERR        | Time reference not initialised   | No calculation performed                              | XL_CFI_TIME_PROC_PRO<br>C_REF_INIT_ERR       | 5        |
| WARN       | Time out of initialization range   | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_PROC_PRO<br>C_REF_INIT_WARN      | 6        |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_PROC_PRO<br>C_BUL_A_FORMULA_WARN | 7        |
| WARN       | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_PROC_PRO<br>C_BUL_B_A_GAP_WARN   | 8        |
| WARN       | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_PROC_PRO<br>C_VALIDITY_WARN      | 9        |



## 7.15 **xl\_time\_processing\_to\_transport**

### 7.15.1 **Overview**

The **xl\_time\_processing\_to\_transport** CFI function transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.15.2 **Calling Interface**

The calling interface of the **xl\_time\_processing\_to\_transport** CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long proc_id_in, trans_id_out;
    long time_ref_in, time_ref_out;
    double processing_in;
    long transport_out[XL_TIME_TRANS_DIM_MAX];
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_PROC_TRANS], status;

    status = xl_time_processing_to_transport(&time_id, &proc_id_in,
        &time_ref_in, &processing_in, &trans_id_out,
        &time_ref_out, transport_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_processing_to_transport_run(&run_id, &proc_id_in,
        &time_ref_in, &processing_in, &trans_id_out,
        &time_ref_out, transport_out, ierr);
}
```

The **XL\_TIME\_TRANS\_DIM\_MAX** and **XL\_NUM\_ERR\_PROC\_TRANS** constants are defined in the file *explorer\_lib.h*.

### 7.15.3 Input Parameters

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

*Table 49: Input parameters of `xl_time_processing_to_transport` function*

| C name                     | C type                   | Array Element | Description (Reference)                        | Unit (Format)                      | Allowed Range                         |
|----------------------------|--------------------------|---------------|--|------------------------------------|---------------------------------------|
| <code>time_id</code>       | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -                                  | -                                     |
| <code>proc_id_in</code>    | <code>long *</code>      | -             | Processing format ID                           | -                                  | Complete                              |
| <code>time_ref_in</code>   | <code>long *</code>      | -             | Time reference ID                              | -                                  | Any except <code>XL_TIME_UNDEF</code> |
| <code>processing_in</code> | <code>double*</code>     | -             | Time in Processing Format                      | Decimal days, MJD2000 (Processing) | [-18262.0,36524.0]                    |
| <code>trans_id_out</code>  | <code>long *</code>      | -             | Transport format ID                            | -                                  | Complete                              |
| <code>time_ref_out</code>  | <code>long *</code>      | -             | Time reference ID                              | -                                  | Any except <code>XL_TIME_UNDEF</code> |

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

- Processing format ID: `proc_id_in`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].
- Transport format ID: `trans_id_out`. Current document, section 6.2.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.15.4 Output Parameters

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

*Table 50: Output parameters of `xl_time_processing_to_transport`*

| C name                                       | C type            | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|--|-------------------|---------------|--------------------------|---------------|---------------|
| <code>xl_time_processing_to_transport</code> | <code>long</code> | -             | Status flag              | -             | -             |
| <code>transport_out[dim]</code>              | <code>long</code> | See Table 3   | Time in Transport format | See Table 3   | See Table 3   |
| <code>ierr</code>                            | <code>long</code> | -             | Error vector             | -             | -             |

### 7.15.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl\_time\_processing\_to\_transport** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 **xl\_time\_processing\_to\_transport** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM])

**Table 51: Error messages of xl\_time\_processing\_to\_transport function**

| Error type | Error message  | Cause and impact                                   | Error code                                | Error No |
|------------|--|--|---|----------|
| ERR        | Input processing format ID is not correct  | No calculation performed                           | XL_CFI_TIME_PROC_TRANS_PROC_IN_ERR        | 0        |
| ERR        | Input time reference ID is not correct   | No calculation performed                           | XL_CFI_TIME_PROC_TRANS_TIME_IN_ERR        | 1        |
| ERR        | Output transport format ID is not correct  | No calculation performed                           | XL_CFI_TIME_PROC_TRANS_TRANS_OUT_ERR      | 2        |
| ERR        | Output time reference ID is not correct  | No calculation performed                           | XL_CFI_TIME_PROC_TRANS_TIME_OUT_ERR       | 3        |
| ERR        | Number of days out of range  | No calculation performed                           | XL_CFI_TIME_PROC_TRANS_DAY_ERR            | 4        |
| ERR        | Time reference not initialised   | No calculation performed                           | XL_CFI_TIME_PROC_TRANS_REF_INIT_ERR       | 5        |
| WARN       | Time out of initialization range   | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_TRANS_REF_INIT_WARN      | 6        |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_TRANS_BUL_A_FORMULA_WARN | 7        |
| WARN       | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_TRANS_BUL_B_A_GAP_WARN   | 8        |
| WARN       | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_PROC_TRANS_VALIDITY_WARN      | 9        |

## 7.16xl\_time\_transport\_to\_ascii

### 7.16.1Overview

The `xl_time_transport_to_ascii` CFI function transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

### 7.16.2Calling Interface

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

```
#include <explorer_lib.h>
{
    long trans_id_in, ascii_id_out;
    long time_ref_in, time_ref_out;
    long transport_in[XL_TIME_TRANS_DIM_MAX];
    char ascii_out[XL_TIME_ASCII_DIM_MAX];
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_TRANS_ASCII], status;

    status = xl_time_transport_to_ascii(&time_id, &trans_id_in,
                                       &time_ref_in, transport_in, &ascii_id_out,
                                       &time_ref_out, ascii_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_transport_to_ascii_run(&run_id, &trans_id_in,
                                           &time_ref_in, transport_in, &ascii_id_out,
                                           &time_ref_out, ascii_out, ierr);
}
```

The `XL_TIME_TRANS_DIM_MAX`, `XL_TIME_ASCII_DIM_MAX`, `XL_NUM_ERR_TRANS_ASCII` constants are defined in the file `explorer_lib.h`.

### 7.16.3 Input Parameters

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

*Table 52: Input parameters of `xl_time_transport_to_ascii` function*

| C name                         | C type                   | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range                         |
|--------------------------------|--------------------------|---------------|--|---------------|---------------------------------------|
| <code>time_id</code>           | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -             | -                                     |
| <code>trans_id_in</code>       | <code>long *</code>      | -             | Transport format ID                            | -             | Complete                              |
| <code>time_ref_in</code>       | <code>long *</code>      | -             | Time reference ID                              | -             | Any except <code>XL_TIME_UNDEF</code> |
| <code>transport_in[dim]</code> | <code>long</code>        | See tTable 3  | Time in Transport format                       | See Table 3   | See Table 3                           |
| <code>ascii_id_out</code>      | <code>long *</code>      | -             | ASCII format ID                                | -             | Complete                              |
| <code>time_ref_out</code>      | <code>long *</code>      | -             | Time reference ID                              | -             | Any except <code>XL_TIME_UNDEF</code> |

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

- Transport format ID: `trans_id_in`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].
- ASCII format ID: `ascii_id_out`. Current document, section 6.2.

It is important to point out the usage of the `time_ref_out` parameter within the current function:

- If the time reference flag for the output is undefined, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.16.4 Output Parameters

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

*Table 53: Output parameters of `xl_time_transport_to_ascii`*

| C name                                  | C type            | Array Element       | Description (Reference) | Unit (Format)           | Allowed Range           |
|---|-------------------|---------------------|-------------------------|-------------------------|-------------------------|
| <code>xl_time_transport_to_ascii</code> | <code>long</code> | -                   | Status flag             | -                       | -                       |
| <code>ascii_out</code>                  | <code>char</code> | See Table 4 Table 5 | Time in ASCII format    | See Table 4 and Table 5 | See Table 4 and Table 5 |
| <code>ierr</code>                       | <code>long</code> | -                   | Error vector            | -                       | -                       |

### 7.16.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_transport_to_ascii` CFI

function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_transport_to_ascii` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 54: Error messages of `xl_time_transport_to_ascii` function**

| Error type | Error message  | Cause and impact                                      | Error code   | Error No |
|------------|--|---|--|----------|
| ERR        | Input transport format ID is not correct                   | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_TRANS_IN_ERR           | 0        |
| ERR        | Input time reference ID is not correct                     | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_TIME_IN_ERR            | 1        |
| ERR        | Number of days out of range                                | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_DAY_ERR                | 2        |
| ERR        | Number of seconds out of range                             | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_SEC_ERR                | 3        |
| ERR        | Number of milliseconds out of range                        | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_MILLISEC_ERR           | 4        |
| ERR        | Number of microseconds out of range                        | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_MICROSEC_ERR           | 5        |
| ERR        | Number of SIRAL extra counter ticks out of range           | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_TICK_ERR               | 6        |
| ERR        | Output ascii format ID is not correct                      | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_OUT_ERR          | 7        |
| ERR        | Output time reference ID is not correct                    | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_TIME_OUT_ERR           | 8        |
| ERR        | Internal error: Output ascii MJD is out of range           | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_MJD_ERR          | 9        |
| ERR        | Internal error: Output ascii year is out of range          | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_YEAR_ERR         | 10       |
| ERR        | Internal error: Output ascii month is out of range         | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_MONTH_ERR        | 11       |
| ERR        | Internal error: Output ascii day is out of range           | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_DAY_ERR          | 12       |
| ERR        | Internal error: Output ascii hour is out of range          | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_HOUR_ERR         | 13       |
| ERR        | Internal error: Output ascii minutes are out of range      | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_MIN_ERR          | 14       |
| ERR        | Internal error: Output ascii seconds are out of range      | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_SEC_ERR          | 15       |
| ERR        | Internal error: Output ascii microseconds are out of range | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_ASCII_MICROSEC_ER<br>R | 16       |
| ERR        | Internal error: Output ascii format is not correct         | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_FORMAT_OUT_ERR         | 17       |
| ERR        | Time reference not initialised                             | No calculation performed                              | XL_CFI_TIME_TRANS_AS<br>CII_REF_INIT_ERR           | 18       |
| WARN       | Time out of initialization range                           | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_TRANS_AS<br>CII_REF_INIT_WARN          | 19       |

|      |  |   |  |    |
|------|--|---|--|----|
| WARN | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_TRANS_ASC<br>II_BUL_A_FORMULA_WARN | 20 |
| WARN | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_TRANS_ASC<br>II_BUL_B_A_GAP_WARN   | 21 |
| WARN | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed.<br>A message informs the user. | XL_CFI_TIME_TRANS_ASC<br>II_VALIDITY_WARN      | 22 |

## 7.17 **xl\_time\_transport\_to\_processing**

### 7.17.1 **Overview**

The **xl\_time\_transport\_to\_processing** CFI function transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.17.2 **Calling Interface**

The calling interface of the **xl\_time\_transport\_to\_processing** CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long trans_id_in, proc_id_out;
    long time_ref_in, time_ref_out;
    long transport_in[XL_TIME_TRANS_DIM_MAX];
    double processing_out;
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_TRANS_PROC], status;

    status = xl_time_transport_to_processing(&time_id,
&trans_id_in,
                                     &time_ref_in, transport_in, &proc_id_out,
                                     &time_ref_out, &processing_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_transport_to_processing_run(&run_id, &trans_id_in,
                                     &time_ref_in, transport_in, &proc_id_out,
                                     &time_ref_out, &processing_out, ierr);
}
```

The `XL_TIME_TRANS_DIM_MAX` and `XL_NUM_ERR_TRANS_PROC` constants are defined in the file *explorer\_lib.h*.



### 7.17.3 Input Parameters

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

*Table 55: Input parameters of `xl_time_transport_to_processing` function*

| C name                         | C type                   | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range            |
|--------------------------------|--------------------------|---------------|--|---------------|--------------------------|
| <code>time_id</code>           | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -             | -                        |
| <code>trans_id_in</code>       | <code>long *</code>      | -             | Transport format ID                            | -             | Complete                 |
| <code>time_ref_in</code>       | <code>long *</code>      | -             | Time reference ID                              | -             | Any except XL_TIME_UNDEF |
| <code>transport_in[dim]</code> | <code>long</code>        | See Table 3   | Time in Transport format                       | See Table 3   | See Table 3              |
| <code>proc_id_out</code>       | <code>long *</code>      | -             | Processing format ID                           | -             | Complete                 |
| <code>time_ref_out</code>      | <code>long *</code>      | -             | Time reference ID                              | -             | Any except XL_TIME_UNDEF |

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

- Transport format ID: `trans_id_in`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].
- Processing format ID: `proc_id_out`. Current document, section 6.2

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.17.4 Output Parameters

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

*Table 56: Output parameters of `xl_time_transport_to_processing`*

| C name                                       | C type               | Array Element | Description (Reference)   | Unit (Format)                      | Allowed Range      |
|--|----------------------|---------------|---------------------------|------------------------------------|--------------------|
| <code>xl_time_transport_to_processing</code> | <code>long</code>    | -             | Status flag               | -                                  | -                  |
| <code>processing_out</code>                  | <code>double*</code> | -             | Time in Processing Format | Decimal days, MJD2000 (Processing) | [-18262.0,36524.0] |
| <code>ierr</code>                            | <code>long</code>    | -             | Error vector              | -                                  | -                  |

### 7.17.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl\_time\_transport\_to\_processing** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 by translating the error vector returned by the **xl\_time\_transport\_to\_processing** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM])

**Table 57: Error messages of xl\_time\_transport\_to\_processing function**

| Error type | Error message  | Cause and impact                                   | Error code                                | Error No |
|------------|--|--|---|----------|
| ERR        | Input transport format ID is not correct   | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_TRANS_IN_ERR       | 0        |
| ERR        | Input time reference ID is not correct   | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_TIME_IN_ERR        | 1        |
| ERR        | Output processing format ID is not correct   | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_PROC_OUT_ERR       | 2        |
| ERR        | Output time reference ID is not correct  | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_TIME_OUT_ERR       | 3        |
| ERR        | Number of days out of range  | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_DAY_ERR            | 4        |
| ERR        | Number of seconds out of range   | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_SEC_ERR            | 5        |
| ERR        | Number of milliseconds out of range  | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_MILLISEC_ERR       | 6        |
| ERR        | Number of microseconds out of range  | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_MICROSEC_ERR       | 7        |
| ERR        | Number of SIRAL extra counter ticks out of range   | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_TICK_ERR           | 8        |
| ERR        | Time reference not initialised   | No calculation performed                           | XL_CFI_TIME_TRANS_PROC_REF_INIT_ERR       | 9        |
| WARN       | Time out of initialization range   | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_PROC_REF_INIT_WARN      | 10       |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_PROC_BUL_A_FORMULA_WARN | 11       |
| WARN       | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_PROC_BUL_B_A_GAP_WARN   | 12       |
| WARN       | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_PROC_VALIDITY_WARN      | 13       |



Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 107

---

## 7.18xl\_time\_transport\_to\_transport

### 7.18.1Overview

The `xl_time_transport_to_transport` CFI function transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in a different Transport format and/or reference (TAI, UTC, UT1 or GPS).

### 7.18.2Calling Interface

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

```
#include <explorer_lib.h>
{
    long trans_id_in, trans_id_out;
    long time_ref_in, time_ref_out;
    long transport_in[XL_TIME_TRANS_DIM_MAX];
    long transport_out[XL_TIME_TRANS_DIM_MAX];
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_TRANS_TRANS], status;

    status = xl_time_transport_to_transport(&time_id, &trans_id_in,
                                           &time_ref_in, transport_in, &trans_id_out,
                                           &time_ref_out, transport_out, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_time_transport_to_transport_run(&run_id, &trans_id_in,
                                               &time_ref_in, transport_in, &trans_id_out,
                                               &time_ref_out, transport_out, ierr);
}
```

### 7.18.3Input Parameters

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

**Table 58: Input parameters of `xl_time_transport_to_transport` function**

| C name                         | C type                   | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range                         |
|--------------------------------|--------------------------|---------------|--|---------------|---------------------------------------|
| <code>time_id</code>           | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -             | -                                     |
| <code>trans_id_in</code>       | <code>long *</code>      | -             | Transport format ID                            | -             | Complete                              |
| <code>time_ref_in</code>       | <code>long *</code>      | -             | Time reference ID                              | -             | Any except <code>XL_TIME_UNDEF</code> |
| <code>transport_in[dim]</code> | <code>long</code>        | See Table 3   | Time in Transport format                       | See Table 3   | See Table 3                           |
| <code>trans_id_out</code>      | <code>long *</code>      | -             | Transport format ID                            | -             | Complete                              |
| <code>time_ref_out</code>      | <code>long *</code>      | -             | Time reference ID                              | -             | Any except <code>XL_TIME_UNDEF</code> |

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

- Transport format ID: `trans_id_in` and `trans_id_out`. Current document, section 6.2.
- Time reference ID: `time_ref_in` and `time_ref_out`. See [GEN\_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless `time_ref_in = time_ref_out`.

### 7.18.4 Output Parameters

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

**Table 59: Output parameters of `xl_time_transport_to_transport`**

| C name                                      | C type            | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|---|-------------------|---------------|--------------------------|---------------|---------------|
| <code>xl_time_transport_to_transport</code> | <code>long</code> | -             | Status flag              | -             | -             |
| <code>transport_out[dim]</code>             | <code>long</code> | See Table 3   | Time in Transport format | See Table 3   | See Table 3   |
| <code>ierr</code>                           | <code>long</code> | -             | Error vector             | -             | -             |

### 7.18.5 Warnings and Errors

Next table lists the possible error messages that can be returned by the `xl_time_transport_to_transport` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_time_transport_to_transport` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 60: Error messages of xl\_time\_transport\_to\_transport function**

| Error type | Error message  | Cause and impact                                   | Error code                                     | Error No |
|------------|--|--|--|----------|
| ERR        | Input transport format ID is not correct   | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_IN_ERR             | 0        |
| ERR        | Input time reference ID is not correct   | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_TIME_IN_ERR        | 1        |
| ERR        | Output transport format ID is not correct  | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_OUT_ERR            | 2        |
| ERR        | Output time reference ID is not correct  | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_TIME_OUT_ERR       | 3        |
| ERR        | Number of days out of range  | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_DAY_ERR            | 4        |
| ERR        | Number of seconds out of range   | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_SEC_ERR            | 5        |
| ERR        | Number of milliseconds out of range  | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_MILLISEC_ERR       | 6        |
| ERR        | Number of microseconds out of range  | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_MICROSEC_ERR       | 7        |
| ERR        | Number of SIRAL extra counter ticks out of range   | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_TICK_ERR           | 8        |
| ERR        | Time reference not initialised   | No calculation performed                           | XL_CFI_TIME_TRANS_TRANSPORT_REF_INIT_ERR       | 9        |
| WARN       | Time out of initialization range   | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_TRANSPORT_REF_INIT_WARN      | 10       |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_TRANSPORT_BUL_A_FORMULA_WARN | 11       |
| WARN       | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_TRANSPORT_BUL_B_A_GAP_WARN   | 12       |
| WARN       | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed. A message informs the user. | XL_CFI_TIME_TRANS_TRANSPORT_VALIDITY_WARN      | 13       |

## 7.19 **xl\_time\_add**

### 7.19.1 Overview

The **xl\_time\_add** CFI function adds a time duration to a TAI, UTC, UT1 or GPS times expressed in Processing format.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.19.2 Calling interface

The calling interface of the **xl\_time\_add** CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long proc_id, time_ref;
    double processing_in, added_duration, processing_out;
    long ierr[XL_NUM_ERR_TIME_ADD], status;

    status = xl_time_add (&proc_id, &time_ref,
                        &processing_in, &added_duration,
                        &processing_out, ierr);
}
```

The XL\_NUM\_ERR\_TIME\_ADD constant is defined in the file *explorer\_lib.h*.

### 7.19.3 Input parameters

The **xl\_time\_add** CFI function has the following input parameters:

**Table 61: Input parameters of xl\_time\_add function**

| C name         | C type  | Array Element | Description (Reference)   | Unit (Format)                             | Allowed Range            |
|----------------|---------|---------------|---------------------------|---|--------------------------|
| proc_id        | long *  | -             | Processing format ID      | -   | Complete                 |
| time_ref       | long *  | -             | Time reference ID         | -   | Any except XL_TIME_UNDEF |
| processing_in  | double* | -             | Time in Processing Format | Decimal days, MJD2000 (Processing format) | [-18262.0,36524.0]       |
| added_duration | double* | -             | Duration to be added      | Decimal days (Processing format)          | -                        |

It is important to point out that the duration is not a time, but a time interval expressed in decimal days to be added to the original time.

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

- Processing format ID: `proc_id`. Current document, section 6.2.
- Time reference ID: `time_ref`. See [GEN\_SUM].

### 7.19.4 Output parameters

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

**Table 62: Output parameters of `xl_time_add` function**

| C name                         | C type  | Array Element | Description (Reference)   | Unit (Format)                    | Allowed Range      |
|--------------------------------|---------|---------------|---------------------------|----------------------------------|--------------------|
| <code>xl_time_add</code>       | long    | -             | Status flag               | -                                | -                  |
| <code>processing_output</code> | double* | -             | Time in Processing Format | Decimal days (Processing format) | [-18262.0,36524.0] |
| <code>ierr</code>              | long    | -             | Error vector              | -                                | -                  |

### 7.19.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_time_add` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_add` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 63: Error messages of `xl_time_add` function**

| Error type | Error message                          | Cause and impact         | Error code                  | Error No |
|------------|--|--------------------------|-----------------------------|----------|
| ERR        | Processing format ID is not correct    | No calculation performed | XL_CFI_TIME_ADD_PROC_ERR    | 0        |
| ERR        | Time reference ID is not correct       | No calculation performed | XL_CFI_TIME_ADD_TIME_ERR    | 1        |
| ERR        | Input processing time is out of range  | No calculation performed | XL_CFI_TIME_ADD_DAY_IN_ERR  | 2        |
| ERR        | Output processing time is out of range | No calculation performed | XL_CFI_TIME_ADD_DAY_OUT_ERR | 3        |





Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 113

---

## 7.20 xl\_time\_diff

### 7.20.1 Overview

The `xl_time_diff` CFI function calculates the time difference between two TAI, UTC, UT1 or GPS times expressed in Processing format.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

### 7.20.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long proc_id, time_ref;
    double processing_in_1, processing_in_2, processing_out;
    long ierr[XL_NUM_ERR_TIME_DIFF], status;

    status = xl_time_diff (&proc_id, &time_ref,
                          &processing_in_1, &processing_in_2,
                          &processing_out, ierr);
}
```

Note that `processing_out` is a duration, not a time itself, so it should not be converted to another reference or format.

The `XL_NUM_ERR_TIME_DIFF` constant is defined in the file `explorer_lib.h`.

### 7.20.3 Input parameters

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

**Table 64: Input parameters of `xl_time_diff` function**

| C name                       | C type  | Array Element | Description (Reference)   | Unit (Format)                             | Allowed Range                         |
|------------------------------|---------|---------------|---------------------------|---|---------------------------------------|
| <code>proc_id</code>         | long *  | -             | Processing format ID      | -   | Complete                              |
| <code>time_ref</code>        | long *  | -             | Time reference ID         | -   | Any except <code>XL_TIME_UNDEF</code> |
| <code>processing_in_1</code> | double* | -             | Time in Processing Format | Decimal days, MJD2000 (Processing format) | [-18262.0,36524.0]                    |
| <code>processing_in_2</code> | double* | -             | Time in Processing        | Decimal days,                             | [-18262.0,36524.0]                    |

|  |  |  |        |                                |  |
|--|--|--|--------|--------------------------------|--|
|  |  |  | Format | MJD2000<br>(Processing format) |  |
|--|--|--|--------|--------------------------------|--|

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

- Processing format ID: `proc_id`. Current document, section 6.2.
- Time reference ID: `time_ref`. See [GEN\_SUM].

### 7.20.4 Output parameters

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

**Table 65: Output parameters of `xl_time_diff` function**

| C name                      | C type  | Array Element | Description (Reference)   | Unit (Format)                       | Allowed Range |
|-----------------------------|---------|---------------|---|-------------------------------------|---------------|
| <code>xl_time_diff</code>   | long    | -             | Status flag   | -                                   | -             |
| <code>processing_out</code> | double* | -             | Time difference between <code>processing_in_1</code> and <code>processing_in_2</code> expressed in decimal days | Decimal days<br>(Processing format) | -             |
| <code>ierr</code>           | long    | -             | Error vector  | -                                   | -             |

### 7.20.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_time_diff` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_diff` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 66: Error messages of `xl_time_diff` function**

| Error type | Error message                            | Cause and impact         | Error code                     | Error No |
|------------|--|--------------------------|--------------------------------|----------|
| ERR        | Processing format ID is not correct      | No calculation performed | XL_CFI_TIME_DIFF_PROC_ERR      | 0        |
| ERR        | Time reference ID is not correct         | No calculation performed | XL_CFI_TIME_DIFF_TIME_ERR      | 1        |
| ERR        | Input processing time #1 is out of range | No calculation performed | XL_CFI_TIME_DIFF_DAY_I N_1_ERR | 2        |
| ERR        | Input processing time #2 is out of range | No calculation performed | XL_CFI_TIME_DIFF_DAY_I N_2_ERR | 3        |



Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 116

---

## 7.21xl\_time\_obt\_to\_time

### 7.21.10overview

The `xl_time_obt_to_time` CFI function transforms from On-board Time (OBT) count to UTC Processing time.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

See [MCD] or details on time formats and representations, in particular the definition of OBT.

Note that in the Envisat OBT case there is an ambiguity on the UTC to be computed, because a given OBT count corresponds to many possible times. This is due to the wrap-around of the OBT counter, which occurs about every 190 days.

To solve the ambiguity, the chosen time (given as output) is the time nearest to the reference (given as input) and corresponding to the specified OBT (also given as input).

The `xl_time_obt_to_time` CFI function applies to satellites where OBT time is a counter, which needs to be correlated to an actual time reference. Nevertheless, some other satellites, like Cryosat, use an actual time reference on-board. In this case, the on-board time conversions are handled by the `xl_time_processing_to_processing` function.

Due to the different OBT models used by the various spacecraft, specific data structures are used for each of them. The keep a single interface for the function, a void pointer is used to pass the specific structures to the generic function.

The following data structures are defined for ENVISAT:

```
/* Envisat OBT Structure */
typedef struct
{
    long          sat_id;
    double        time0;
    unsigned long obt0[2];
    unsigned long period0;
} xl_envisat_obt_param;

typedef struct
{
    long          sat_id;
    unsigned long obt[2];
} xl_envisat_obt_value;
```

for GOCE:

```
/* GOCE OBT Structure */
typedef struct
{
```

```
long          sat_id;
unsigned long utc0_c;
unsigned int  utc0_f;
unsigned long obt0_c;
unsigned int  obt0_f;
double       gradient;
double       offset;
} xl_goce_obt_param;
```

```
typedef struct
{
    long          sat_id;
    double       obt;
} xl_goce_obt_value;
```

#### for SMOS

```
typedef struct
{
    long sat_id;
    long delta_seconds; /* number of seconds to be applied to UTC to
                        give UTC Proteus (just in case UTC Proteus
                        reference is actually GPS Time)*/
    unsigned long obet0_c; /* OBET Coarse Time (in seconds) */
    unsigned long obet0_f; /* OBET Fine Time */
    unsigned long utc0_week; /* UTC (Proteus format) week number */
    unsigned long utc0_seconds; /* UTC (Proteus format) seconds of
                                week */
    unsigned long utc0_fraction; /* UTC (Proteus format) fraction of
                                seconds */
} xl_smos_obt_param;
```

```
typedef struct
{
    long sat_id;
    unsigned long obet_c; /* OBET Coarse Time (in seconds) */
    unsigned long obet_f; /* OBET Fine Time */
} xl_smos_obt_value;
```

and for ADM

```
typedef struct
{
    long sat_id;
    long delta_seconds; /* it refers to the number of seconds to be
                        applied to UTC to give GPS (GPST - UTC) */
} xl_adm_obt_param;

typedef struct
{
    long sat_id;
    unsigned long cuc_sec; /* CCSDS Unsegmented Time Code (secs) */
    unsigned long cuc_subsec; /* CCSDS Unsegmented Time Code
                              (subseconds) */
} xl_adm_obt_value;
```

The `sat_id` parameter within the structure has to be assigned equal to the `sat_id` passed to the function.

### 7.21.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long sat_id, proc_id;
    xl_envisat_obt_param obt_param; /*example for ENVISAT */
    xl_envisat_obt_value obt_value_in; /*example for ENVISAT */
    double time_out;
    long ierr[XL_NUM_ERR_OBT_TIME], status;

    status = xl_time_obt_to_time (&sat_id,
                                &proc_id,
                                &obt_param,
                                &obt_value_in,
                                &time_out,
                                ierr);
```

```

/* Or, using the run_id */
long run_id;
status = xl_time_obt_to_time_run (&run_id,
                                  &proc_id,
                                  &obt_param,
                                  &obt_value_in,
                                  &time_out,
                                  err);
}

```

The XL\_NUM\_ERR\_OBT\_TIME constant is defined in the file *explorer\_lib.h*.

### 7.21.3 Input parameters

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

**Table 67: Input parameters of `xl_time_obt_to_time` function**

| C name       | C type | Array Element | Description (Reference)                                | Unit (Format) | Allowed Range |
|--------------|--------|---------------|--|---------------|---------------|
| sat_id       | long * | -             | Satellite ID   | -             | Complete      |
| proc_id      | long * | -             | Processing format ID                                   | -             | Complete      |
| obt_param    | void * | -             | Pointer to <code>xl_&lt;satellite&gt;_obt_param</code> | -             | -             |
| obt_value_in | void * | -             | Pointer to <code>xl_&lt;satellite&gt;_obt_value</code> | -             | -             |

**Table 68: Input parameters of `xl_envisat_obt_param` structure**

| C name  | C type           | Array Element | Description (Reference)  | Unit (Format)                    | Allowed Range      |
|---------|------------------|---------------|--|----------------------------------|--------------------|
| sat_id  | long             | -             | Satellite ID   | -                                | XL_SAT_ENVISAT     |
| time0   | double           | -             | Reference time   | Decimal days (Processing format) | [-18262.0,36524.0] |
| obt0    | unsigned long[2] | -             | Array of counters containing the OBT at the reference time (in the satellite dependant format) | TBD                              | TBD                |
| period0 | unsigned long    | -             | Actual on-board clock period   | TBD                              | TBD                |



**Table 69: Input parameters of xl\_envisat\_obt\_value structure**

| C name | C type           | Array Element | Description (Reference)   | Unit (Format) | Allowed Range  |
|--------|------------------|---------------|---|---------------|----------------|
| sat_id | long             | -             | Satellite ID  | -             | XL_SAT_ENVISAT |
| obt    | unsigned long[2] | -             | Array of counters containing the OBT time (in the satellite dependant format) | TBD           | TBD            |

**Table 70: Input parameters of xl\_goce\_obt\_param structure**

| C name   | C type        | Array Element | Description (Reference)              | Unit (Format)            | Allowed Range |
|----------|---------------|---------------|--------------------------------------|--------------------------|---------------|
| sat_id   | long          | -             | Satellite ID                         | -                        | XL_SAT_GOCE   |
| utc0_c   | unsigned long | -             | Coarse UTC0                          | seconds                  | >=0           |
| utc0_f   | unsigned int  | -             | Fine UTC0                            | 2 <sup>-16</sup> seconds | >=0           |
| obt0_c   | unsigned long | -             | Coarse OBT0                          | seconds                  | >=0           |
| obt0_f   | unsigned int  | -             | Fine OBT0                            | 2 <sup>-16</sup> seconds | >=0           |
| gradient | double        | -             | Gradient between the OBT and the UTC | -                        | -             |
| offset   | double        | -             | Offset between the OBT and the UTC   | seconds                  | -             |

**Table 71: Input parameters of xl\_goce\_obt\_value structure**

| C name | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------|--------|---------------|-------------------------|---------------|---------------|
| sat_id | long   | -             | Satellite ID            | -             | XL_SAT_GOCE   |
| obt    | double | -             | OBT time                | sconds        | -             |

**Table 72: Input parameters of xl\_smos\_obt\_param structure**

| C name        | C type        | Array Element | Description (Reference)   | Unit (Format)            | Allowed Range |
|---------------|---------------|---------------|---|--------------------------|---------------|
| sat_id        | long          | -             | Satellite ID  | -                        | XL_SAT_SMOS   |
| delta_seconds | long          | -             | Number of seconds to be applied to UTC to give UTC Proteus (in case UTC Proteus reference is actually GPS Time) | seconds                  |               |
| obet0_c       | unsigned long | -             | OBETO Coarse Time   | seconds                  | >=0           |
| obet0_f       | unsigned long | -             | OBET 0Fine Time   | 2 <sup>-16</sup> seconds | >=0           |
| utc0_week     | unsigned long | -             | UTC0 (Proteus format) week number   | weeks                    | >=0           |

|               |               |   |   |                          |     |
|---------------|---------------|---|---|--------------------------|-----|
| utc0_second   | unsigned long | - | UTC0 (Proteus format) seconds of week     | seconds                  | >=0 |
| utc0_fraction | unsigned long | - | UTC0 (Proteus format) fraction of seconds | 2 <sup>-16</sup> seconds | >=0 |

**Table 73: Input parameters of xl\_smos\_obt\_value structure**

| C name | C type        | Array Element | Description (Reference) | Unit (Format)            | Allowed Range |
|--------|---------------|---------------|-------------------------|--------------------------|---------------|
| sat_id | long          | -             | Satellite ID            | -                        | XL_SAT_SMOS   |
| obet_c | unsigned long | -             | OBET Coarse Time        | seconds                  | >=0           |
| obet_f | unsigned long | -             | OBET Fine Time          | 2 <sup>-16</sup> seconds | >=0           |

**Table 74: Input parameters of xl\_adm\_obt\_param structure**

| C name        | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|---------------|--------|---------------|---|---------------|---------------|
| sat_id        | long   | -             | Satellite ID  | -             | XL_SAT_ADM    |
| delta_seconds | long   | -             | Number of seconds to be applied to UTC to give GPS (GPST - UTC) | seconds       |               |

**Table 75: Input parameters of xl\_adm\_obt\_value structure**

| C name     | C type        | Array Element | Description (Reference)                  | Unit (Format)            | Allowed Range |
|------------|---------------|---------------|--|--------------------------|---------------|
| sat_id     | long          | -             | Satellite ID                             | -                        | XL_SAT_ADM    |
| cuc_sec    | unsigned long | -             | CCSDS Unsegmented Time Code (seconds)    | seconds                  | >=0           |
| cuc_subsec | unsigned long | -             | CCSDS Unsegmented Time Code (subseconds) | 2 <sup>-16</sup> seconds | >=0           |

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

- Satellite ID: sat\_id. See [GEN\_SUM].
- Processing format ID: proc\_id. Current document, section 6.2.

### **7.21.4 Output parameters**

The output parameters of the xl\_time\_obt\_to\_time CFI function are:

**Table 76: Output parameters of `xl_time_obt_to_time` function**

| C name                           | C type  | Array Element | Description (Reference)       | Unit (Format)                             | Allowed Range      |
|----------------------------------|---------|---------------|-------------------------------|---|--------------------|
| <code>xl_time_obt_to_time</code> | long    | -             | Status flag                   | -   | -                  |
| <code>time_out</code>            | double* | -             | UTC Time in Processing Format | Decimal days, MJD2000 (Processing format) | [-18262.0,36524.0] |
| <code>ierr</code>                | long    | -             | Error vector                  | -   | -                  |

### 7.21.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_time_obt_to_time` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_obt_to_time` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 77: Error messages of `xl_time_obt_to_time` function**

| Error type | Error message                                 | Cause and impact         | Error code                                      | Error No |
|------------|---|--------------------------|---|----------|
| ERR        | Satellite ID is not correct                   | No calculation performed | XL_CFI_TIME_OBT_TIME_SAT_ERR                    | 0        |
| ERR        | Processing format ID is not correct           | No calculation performed | XL_CFI_TIME_OBT_TIME_PROC_ERR                   | 1        |
| ERR        | Structure inconsistent with Satellite ID      | No calculation performed | XL_CFI_TIME_OBT_TIME_INCONSISTENT_STRUCTURE_ERR | 2        |
| ERR        | Input reference time is out of range          | No calculation performed | XL_CFI_TIME_OBT_TIME_DAY_REF_ERR                | 3        |
| ERR        | No OBT defined for this satellite ID          | No calculation performed | XL_CFI_TIME_OBT_TIME_OBT_SAT_ERR                | 4        |
| ERR        | OBT at reference time is out of allowed range | No calculation performed | XL_CFI_TIME_OBT_TIME_OBT_ERR                    | 5        |
| ERR        | Period of the On-Board clock is null          | No calculation performed | XL_CFI_TIME_OBT_TIME_CLOCK_ERR                  | 6        |
| ERR        | Output time is out of range                   | No calculation performed | XL_CFI_TIME_OBT_TIME_DAY_OUT_ERR                | 7        |

**Table 78: Runtime performances of `xl_time_obt_to_time` function**

| Solaris 32-bit. [ms] | Solaris 64 bit. [ms] | Linux 32-bit. [ms] | Linux 64-bit. [ms] |
|----------------------|----------------------|--------------------|--------------------|
| 0.01899              | 0.00254              | 0.00319            | 0.00050            |



Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 124

---

## 7.22xl\_time\_time\_to\_obt

### 7.22.1Overview

The `xl_time_time_to_obt` CFI function transforms a UTC Processing time to OBT count.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

See [MCD] for details on time formats and representations, in particular the definition OBT.

Note that no rounding to any number of significant bits is performed by `xl_time_time_to_obt`. The user application must perform this rounding if necessary. An example of rounding is provided in the example program within the `EO_LIB` library.

The `xl_time_time_to_obt` CFI function applies to satellites where OBT time is a counter, which needs to be correlated to an actual time reference. Nevertheless, some other satellites, like Cryosat, use an actual time reference on-board. In this case, the on-board time conversions are handled by the `xl_time_processing_to_processing` function.

Due to the different OBT models used by the various spacecraft, specific data structures are used for each of them. The keep a single interface for the function, a void pointer is used to pass the specific structures to the generic function.

The following data structures are defined for ENVISAT:

```
/* Envisat OBT Structure */
typedef struct
{
    long          sat_id;
    double        time0;
    unsigned long obt0[2];
    unsigned long period0;
} xl_envisat_obt_param;

typedef struct
{
    long          sat_id;
    unsigned long obt[2];
} xl_envisat_obt_value;
```

for GOCE:

```
/* GOCE OBT Structure */
typedef struct
{
    long          sat_id;
    unsigned long utc0_c;
    unsigned int  utc0_f;
```

```
unsigned long obt0_c;  
unsigned int  obt0_f;  
double       gradient;  
double       offset;  
} xl_goce_obt_param;
```

```
typedef struct  
{  
    long          sat_id;  
    double        obt;  
} xl_goce_obt_value;
```

#### for SMOS

```
typedef struct  
{  
    long sat_id;  
    long delta_seconds; /* number of seconds to be applied to UTC to  
                        give UTC Proteus (just in case UTC Proteus  
                        reference is actually GPS Time)*/  
    unsigned long obet0_c; /* OBET Coarse Time (in seconds) */  
    unsigned long obet0_f; /* OBET Fine Time */  
    unsigned long utc0_week; /* UTC (Proteus format) week number */  
    unsigned long utc0_seconds; /* UTC (Proteus format) seconds of  
                                week */  
    unsigned long utc0_fraction; /* UTC (Proteus format) fraction of  
                                seconds */  
} xl_smos_obt_param;
```

```
typedef struct  
{  
    long sat_id;  
    unsigned long obet_c; /* OBET Coarse Time (in seconds) */  
    unsigned long obet_f; /* OBET Fine Time */  
} xl_smos_obt_value;
```

#### and for ADM

```
typedef struct
```

```
{
    long sat_id;
    long delta_seconds; /* it refers to the number of seconds to be
                        applied to UTC to give GPS (GPST - UTC) */
} xl_adm_obt_param;

typedef struct
{
    long sat_id;
    unsigned long cuc_sec; /* CCSDS Unsegmented Time Code (secs) */
    unsigned long cuc_subsec; /* CCSDS Unsegmented Time Code
                              (subseconds) */
} xl_adm_obt_value;
```

The `sat_id` parameter within the structure has to be assigned equal to the `sat_id` passed to the function.

### **7.22.2 Calling interface**

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

```
#include <explorer_lib.h>
{
    long sat_id, proc_id;
    double time_in;
    xl_envisat_obt_param obt_param; /*example for ENVISAT */
    xl_envisat_obt_value obt_value_out; /*example for ENVISAT */
    long ierr[XL_NUM_ERR_TIME_OBT], status;

    status = xl_time_time_to_obt (&sat_id,
                                &proc_id,
                                &obt_param,
                                &time_in,
                                &obt_value_out,
                                ierr);
```

```

/* Or, using the run_id */
long run_id;

status = xl_time_time_to_obt_run (&run_id,
                                  &proc_id,
                                  &obt_param,
                                  &time_in,
                                  &obt_value_out,
                                  ierr);
}

```

The XL\_NUM\_ERR\_TIME\_OBT constant is defined in the file *explorer\_lib.h*.

### 7.22.3 Input parameters

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

**Table 79: Input parameters of `xl_time_obt_to_time` function**

| C name    | C type  | Array Element | Description (Reference)                                | Unit (Format)                    | Allowed Range      |
|-----------|---------|---------------|--|----------------------------------|--------------------|
| sat_id    | long *  | -             | Satellite ID   | -                                | Complete           |
| proc_id   | long *  | -             | Processing format ID                                   | -                                | Complete           |
| obt_param | void *  | -             | Pointer to <code>xl_&lt;satellite&gt;_obt_param</code> | -                                | -                  |
| time_in   | double* | -             | UTC Time   | Decimal days (Processing format) | [-18262.0,36524.0] |

**Table 80: Input parameters of `xl_envisat_obt_param` structure**

| C name  | C type           | Array Element | Description (Reference)  | Unit (Format)                    | Allowed Range      |
|---------|------------------|---------------|--|----------------------------------|--------------------|
| sat_id  | long             | -             | Satellite ID   | -                                | XL_SAT_ENVISAT     |
| time0   | double           | -             | Reference time   | Decimal days (Processing format) | [-18262.0,36524.0] |
| obt0    | unsigned long[2] | -             | Array of counters containing the OBT at the reference time (in the satellite dependant format) | TBD                              | TBD                |
| period0 | unsigned long    | -             | Actual on-board clock period   | TBD                              | TBD                |



**Table 81: Input parameters of xl\_goce\_obt\_param structure**

| C name   | C type        | Array Element | Description (Reference)              | Unit (Format)            | Allowed Range |
|----------|---------------|---------------|--------------------------------------|--------------------------|---------------|
| sat_id   | long          | -             | Satellite ID                         | -                        | XL_SAT_GOCE   |
| utc0_c   | unsigned long | -             | Coarse UTC0                          | seconds                  | >=0           |
| utc0_f   | unsigned int  | -             | Fine UTC0                            | 2 <sup>-16</sup> seconds | >=0           |
| obt0_c   | unsigned long | -             | Coarse OBT0                          | seconds                  | >=0           |
| obt0_f   | unsigned int  | -             | Fine OBT0                            | 2 <sup>-16</sup> seconds | >=0           |
| gradient | double        | -             | Gradient between the OBT and the UTC | -                        | -             |
| offset   | double        | -             | Offset between the OBT and the UTC   | seconds                  | -             |

**Table 82: Input parameters of xl\_smos\_obt\_param structure**

| C name        | C type        | Array Element | Description (Reference)   | Unit (Format)            | Allowed Range |
|---------------|---------------|---------------|---|--------------------------|---------------|
| sat_id        | long          | -             | Satellite ID  | -                        | XL_SAT_SMOS   |
| delta_seconds | long          | -             | Number of seconds to be applied to UTC to give UTC Proteus (in case UTC Proteus reference is actually GPS Time) | seconds                  |               |
| obet0_c       | unsigned long | -             | OBET0 Coarse Time   | seconds                  | >=0           |
| obet0_f       | unsigned long | -             | OBET0 Fine Time   | 2 <sup>-16</sup> seconds | >=0           |
| utc0_week     | unsigned long | -             | UTC0 (Proteus format) week number   | weeks                    | >=0           |
| utc0_second   | unsigned long | -             | UTC0 (Proteus format) seconds of week   | seconds                  | >=0           |
| utc0_fraction | unsigned long | -             | UTC0 (Proteus format) fraction of seconds   | 2 <sup>-16</sup> seconds | >=0           |

**Table 83: Input parameters of xl\_adm\_obt\_param structure**

| C name        | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|---------------|--------|---------------|---|---------------|---------------|
| sat_id        | long   | -             | Satellite ID  | -             | XL_SAT_ADM    |
| delta_seconds | long   | -             | Number of seconds to be applied to UTC to give GPS (GPST - UTC) | seconds       |               |

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

- Satellite ID: `sat_id`. See [GEN\_SUM].
- Time reference ID: `time_ref`. See [GEN\_SUM].

### 7.22.4 Output parameters

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

**Table 84: Output parameters of `xl_time_time_to_obt` function**

| C name                           | C type | Array Element | Description (Reference)                                | Unit (Format) | Allowed Range |
|----------------------------------|--------|---------------|--|---------------|---------------|
| <code>xl_time_time_to_obt</code> | long   | -             | Status flag  | -             | -             |
| <code>obt_value_out</code>       | void * | -             | Pointer to <code>xl &lt;satellite&gt;_obt_value</code> | -             | -             |
| <code>ierr</code>                | long   | -             | Error vector   | -             | -             |

**Table 85: Output parameters of `xl_envisat_obt_value` structure**

| C name              | C type           | Array Element | Description (Reference)  | Unit (Format) | Allowed Range  |
|---------------------|------------------|---------------|--|---------------|----------------|
| <code>sat_id</code> | long             | -             | Satellite ID   | -             | XL_SAT_ENVISAT |
| <code>obt</code>    | unsigned long[2] | -             | Array of counters containing the OBET time (in the satellite dependant format) | TBD           | TBD            |

**Table 86: Output parameters of `xl_goce_obt_value` structure**

| C name              | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|---------------------|--------|---------------|-------------------------|---------------|---------------|
| <code>sat_id</code> | long   | -             | Satellite ID            | -             | XL_SAT_GOCE   |
| <code>obt</code>    | double | -             | OBET time               | seconds       | -             |

**Table 87: Output parameters of `xl_smos_obt_value` structure**

| C name              | C type        | Array Element | Description (Reference) | Unit (Format)            | Allowed Range |
|---------------------|---------------|---------------|-------------------------|--------------------------|---------------|
| <code>sat_id</code> | long          | -             | Satellite ID            | -                        | XL_SAT_SMOS   |
| <code>obet_c</code> | unsigned long | -             | OBET Coarse Time        | seconds                  | >=0           |
| <code>obet_f</code> | unsigned long | -             | OBET Fine Time          | 2 <sup>-16</sup> seconds | >=0           |

**Table 88: Output parameters of xl\_adm\_obt\_value structure**

| C name     | C type        | Array Element | Description (Reference)                  | Unit (Format)            | Allowed Range |
|------------|---------------|---------------|--|--------------------------|---------------|
| sat_id     | long          | -             | Satellite ID                             | -                        | XL_SAT_ADM    |
| cuc_sec    | unsigned long | -             | CCSDS Unsegmented Time Code (seconds)    | seconds                  | >=0           |
| cuc_subsec | unsigned long | -             | CCSDS Unsegmented Time Code (subseconds) | 2 <sup>-16</sup> seconds | >=0           |

### 7.22.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_time\_time\_to\_obt** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 by translating the error vector returned by the **xl\_time\_time\_to\_obt** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 89: Error messages of xl\_time\_time\_to\_obt function**

| Error type | Error message                                 | Cause and impact         | Error code                                   | Error No |
|------------|---|--------------------------|--|----------|
| ERR        | Satellite ID is not correct                   | No calculation performed | XL_CFI_TIME_TIME_OBT_SAT_ERR                 | 0        |
| ERR        | Processing format ID is not correct           | No calculation performed | XL_CFI_TIME_TIME_OBT_PROC_ERR                | 1        |
| ERR        | Structure inconsistent with Satellite ID      | No calculation performed | XL_CFI_TIME_TIME_OBT_INCONSISTENT_STRUCT_ERR | 2        |
| ERR        | Input time is out of range                    | No calculation performed | XL_CFI_TIME_TIME_OBT_DAY_IN_ERR              | 3        |
| ERR        | Input reference time is out of range          | No calculation performed | XL_CFI_TIME_TIME_OBT_DAY_REF_ERR             | 4        |
| ERR        | No OBT defined for this satellite ID          | No calculation performed | XL_CFI_TIME_TIME_OBT_OBT_SAT_ERR             | 5        |
| ERR        | OBT at reference time is out of allowed range | No calculation performed | XL_CFI_TIME_TIME_OBT_OBT_ERR                 | 6        |
| ERR        | Period of the On-Board clock is null          | No calculation performed | XL_CFI_TIME_TIME_OBT_CLOCK_ERR               | 7        |

**Table 90: Runtime performances of xl\_time\_time\_to\_obt function**

| Solaris 32-bit. | Solaris 64 bit. | Linux 32-bit. | Linux 64-bit. |
|-----------------|-----------------|---------------|---------------|
|-----------------|-----------------|---------------|---------------|

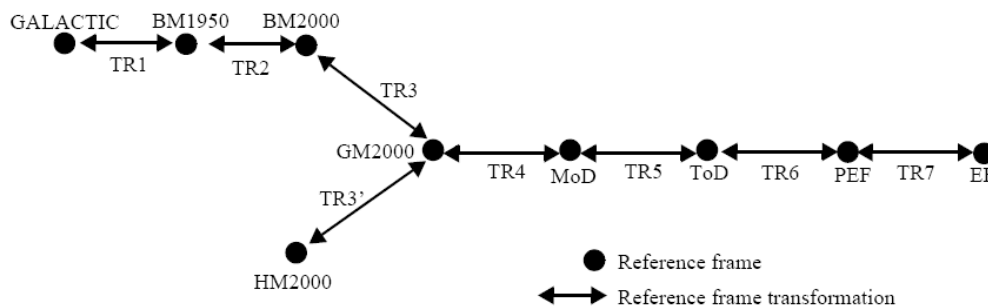
---

| [ms]    | [ms]    | [ms]    | [ms]    |
|---------|---------|---------|---------|
| 0.01904 | 0.00348 | 0.00429 | 0.00063 |

## 7.23xl\_change\_cart\_cs

### 7.23.10 Overview

The `xl_change_cart_cs` CFI function transforms a cartesian state vector between different reference frames. The transformations are done sequentially following the schema in figure . Note that the transformations between BM2000, HM2000 and GM2000 involve a translation of the input vectors (TR3 and TR3'). In case that the input vector is a direction and not a location, the transformation should only apply the rotations between frames, so the transformation should be done in several steps with `xl_change_cart_cs`, skipping the translations<sup>1</sup>.



#### Reference frames:

GALACTIC = Galactic CS (see section 5.1.1)  
 BM1950 = Barycentric Mean of 1950.0 (see section 5.1.2)  
 BM2000 = Barycentric Mean of 2000.0 (see section 5.1.3)  
 HM2000 = Heliocentric Mean of 2000.0 (see section 5.1.4)  
 GM2000 = Geocentric Mean of 2000.0 (see section 5.1.5)  
 MoD = Mean of Date (see section 5.1.6)  
 ToD = True of Date (see section 5.1.7)  
 PEF = Pseudo Earth Fixed (see section 5.1.8)  
 EF = Earth Fixed (see section 5.1.9)

#### Transformations:

TR1 = Galactic to Barycentric Mean of 1950 (see section 5.3.1)  
 TR2 = Barycentric 1950 to Barycentric 2000 (see section 5.3.2)  
 TR3 = Solar system barycentre to Earth centre translation (see section 5.3.3)  
 TR3' = Sun centre to Earth centre translation (see section 5.3.4)  
 TR4 = Precession (see section 5.3.5)  
 TR5 = Nutation (see section 5.3.6)  
 TR6 = Earth rotation + nutation term (see section 5.3.7)  
 TR7 = Polar motion rotation (see section 5.3.8)

**Figure 2: Change cartesian coordinates**

### 7.23.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long mode, cs_in, cs_out, time_ref;
```

<sup>1</sup> For this purpose it is also possible to use the CFI function `xp_change_frame` in the `eo_pointing` library

```

double time;
double pos[3], vel[3], acc[3];
double pos_out[3], vel_out[3], acc_out[3];
xl_model_id model_id = {NULL};
xl_time_id time_id = {NULL};
long status;

status = xl_change_cart_cs (&model_id, &time_id,
                           &mode, &cs_in, &cs_out,
                           &time_ref, &time, pos, vel, acc,
                           pos_out, vel_out, acc_out);

/* Or, using the run_id */
long run_id;

status = xl_change_cart_cs_run (&run_id, &mode, &cs_in,
                                &cs_out,
                                &time_ref, &time, pos, vel, acc,
                                pos_out, vel_out, acc_out);
}

```

### 7.23.3 Input parameters

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

**Table 91: Input parameters of `xl_change_cart_cs` function**

| C name   | C type       | Array Element | Description (Reference)                         | Unit (Format)                    | Allowed Range            |
|----------|--------------|---------------|---|----------------------------------|--------------------------|
| model_id | xl_model_id* | -             | Model ID  | -                                | -                        |
| time_id  | xl_time_id*  | -             | Structure that contains the time correlations.  | -                                | -                        |
| mode     | long*        | -             | Calculation mode selection                      | -                                | Complete                 |
| cs_in    | long *       | -             | Initial reference frame ID                      | -                                | Complete                 |
| cs_out   | long *       | -             | Final reference frame ID                        | -                                | Complete                 |
| time_ref | long *       | -             | Time reference ID                               | -                                | Any except XL_TIME_UNDEF |
| time     | double*      | -             | Reference time                                  | Decimal days (Processing format) | [-18262.0,36524.0]       |
| pos[3]   | double       | all           | Input position vector (Initial reference frame) | m                                | -                        |
| vel[3]   | double       | all           | Input velocity vector                           | m/s                              | -                        |

|                     |        |     |  |         |   |
|---------------------|--------|-----|--|---------|---|
|                     |        |     | (Initial reference frame)<br>This value is dummy if <i>mode</i> is <code>XL_CALC_POS</code> except for the transformations between BM1950 and BM2000   |         |   |
| <code>acc[3]</code> | double | all | Input acceleration vector (Initial reference frame)<br>Dummy if <i>mode</i> is either:<br>· <code>XL_CALC_POS</code><br>· <code>XL_CALC_POS_VEL</code> | $m/s^2$ | - |

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

- Calculation mode selection: `mode`. See current document, section 6.2.
- Time reference ID: `time_ref`. See [GEN\_SUM].
- Reference frame: `cs_in`, `cs_out`. See current document, section 6.2.

Notes:

- the function could not work correctly if the time references are not properly initialised before calling the function (see section 4.2. for details).
- For objects located closer to 1 AU, the transformation from and to BM1950 may produce incorrect results

### 7.23.4 Output parameters

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

**Table 92: Output parameters of `xl_change_cart_cs` function**

| C name                         | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|--------------------------------|--------|---------------|---|---------------|---------------|
| <code>xl_change_cart_cs</code> | long   | -             | Extended status flag  | -             | -             |
| <code>pos_out[3]</code>        | double | all           | Output position vector (Final reference frame)  | m             | -             |
| <code>vel_out[3]</code>        | double | all           | Output velocity vector (Final reference frame)<br>Returned only if <i>mode</i> is either:<br>· <code>XL_CALC_POS_VEL</code><br>· <code>XL_CALC_POS_VEL_ACC</code> | m/s           | -             |
| <code>acc_out[3]</code>        | double | all           | Output acceleration vector (Final reference frame)<br>Returned only if <i>mode</i> is:<br>· <code>XL_CALC_POS_VEL_ACC</code>                                      | $m/s^2$       | -             |

### 7.23.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_change_cart_cs` CFI function

after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 extended status flag returned by the **xl\_change\_cart\_cs** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 93: Error messages of xl\_change\_cart\_cs function**

| Error type | Error message  | Cause and impact         | Error code                             | Error No |
|------------|--|--------------------------|--|----------|
| ERR        | Input time reference ID is not correct   | No calculation performed | XL_CFI_CHANGE_CART_CS_REF_ERR          | 0        |
| ERR        | Input date is out of range   | No calculation performed | XL_CFI_CHANGE_CART_CS_DAY_ERR          | 1        |
| ERR        | Calculation mode ID is not correct   | No calculation performed | XL_CFI_CHANGE_CART_CS_MODE_ERR         | 2        |
| ERR        | Input reference frame is not correct   | No calculation performed | XL_CFI_CHANGE_CART_CS_INPUT_CS_ERR     | 3        |
| ERR        | Output reference frame is not correct  | No calculation performed | XL_CFI_CHANGE_CART_CS_OUTPUT_CS_ERR    | 4        |
| ERR        | Time Reference not initialised   | No calculation performed | XL_CFI_CHANGE_CART_CS_REF_INIT_ERR     | 5        |
| WARN       | Bulletin A: previous computation performed inside file interval, current performed with formula                              | Calculation performed    | XL_CHANGE_CART_CS_BUL_A_FORMULA_WARN   | 6        |
| WARN       | Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.     | Calculation performed    | XL_CHANGE_CART_CS_BUL_B_A_GAP_WARN     | 7        |
| WARN       | Previous computation performed inside initialization validity, current computation performed outside initialization validity | Calculation performed    | XL_CHANGE_CART_CS_VALIDITY_WARN        | 8        |
| ERR        | Error computing TAI time.  | No calculation performed | XL_CHANGE_CART_CS_TIME_COMPUTATION_ERR | 9        |



## 7.24 xl\_geod\_to\_cart

### 7.24.1 Overview

The xl\_geod\_to\_cart CFI function transforms from geodetic to cartesian coordinates.

### 7.24.2 Calling interface

The calling interface of the xl\_geod\_to\_cart CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long mode;
    xl_model_id model_id = {NULL};
    double lon, lat, h, lon_rate, lat_rate, h_rate;
    double pos[3], vel[3];
    long status;

    status = xl_geod_to_cart (&model_id, &mode, &lon, &lat, &h,
                             &lon_rate, &lat_rate, &h_rate,
                             pos, vel);
}
```

### 7.24.3 Input parameters

The xl\_geod\_to\_cart CFI function has the following input parameters:

**Table 94: Input parameters of xl\_geod\_to\_cart function**

| C name   | C type      | Array Element | Description (Reference)                    | Unit (Format) | Allowed Range  |
|----------|-------------|---------------|--|---------------|--|
| model_id | xl_model_id | -             | Model ID                                   | -             | -  |
| mode     | long*       | -             | Calculation mode selection                 | -             | Select either:<br>· XL_CALC_POS<br>· XL_CALC_POS_VEL |
| lon      | double *    | -             | Geocentric longitude (Earth fixed CS)      | deg           | [0,360)  |
| lat      | double *    | -             | Geodetic latitude (Earth fixed CS)         | deg           | [-90,90]   |
| h        | double *    | -             | Geodetic altitude (Earth fixed CS)         | m             | $h \geq -b_{\text{ellipsoid}}$<br>(sat_id dependent) |
| lon_rate | double *    | -             | Geocentric longitude rate (Earth fixed CS) | deg/s         | -  |

|          |          |   |   |       |   |
|----------|----------|---|---|-------|---|
|          |          |   | Dummy if <i>mode</i> is:<br>· XL_CALC_POS   |       |   |
| lat_rate | double * | - | Geodetic latitude rate<br>(Earth fixed CS)<br>Dummy if <i>mode</i> is:<br>· XL_CALC_POS | deg/s | - |
| h_rate   | double * |   | Geodetic altitude rate<br>(Earth fixed CS)<br>Dummy if <i>mode</i> is:<br>· XL_CALC_POS | m/s   | - |

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

- Calculation mode selection: *mode*. See current document, section 6.2.

### 7.24.4 Output parameters

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

**Table 95: Output parameters of `xl_geod_to_cart` function**

| C name                       | C type | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|------------------------------|--------|---------------|--|---------------|---------------|
| <code>xl_geod_to_cart</code> | long   | -             | Extended status flag   | -             | -             |
| <code>pos[3]</code>          | double | all           | Cartesian position vector<br>(Earth fixed CS)  | m             | -             |
| <code>vel[3]</code>          | double | all           | Cartesian velocity vector<br>(Earth fixed CS)<br>Returned only if <i>mode</i> is:<br>· XL_CALC_POS_VEL | m/s           | -             |

### 7.24.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_geod_to_cart` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_geod_to_cart` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 96: Error messages of `xl_geod_to_cart` function**

| Error type | Error message                                   | Cause and impact                                      | Error code                    | Error No |
|------------|---|---|-------------------------------|----------|
| ERR        | Wrong geodetic latitude on input (out of range) | No calculation performed                              | XL_CFI_GEOD_CART_EL_GT_90_ERR | 0        |
| WARN       | Calculation mode ID is not correct              | Calculation performed.<br>A message informs the user. | XL_CFI_GEOD_CART_MODE_WARN    | 1        |

The altitude of the geodetic state vector is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see section 10).

## 7.25 `xl_cart_to_geod`

### 7.25.1 Overview

The `xl_cart_to_geod` CFI function transforms from cartesian to geodetic coordinates.

### 7.25.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long mode;
    xl_model_id model_id = {NULL};
    double pos[3], vel[3];
    double lon, lat, h, lon_rate, lat_rate, h_rate;
    long status;

    status = xl_cart_to_geod (&model_id, &mode, pos, vel, &lon,
    &lat, &h, &lon_rate, &lat_rate, &h_rate);
}
```

### 7.25.3 Input parameters

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

**Table 97: Input parameters of `xl_cart_to_geod` function**

| C name                | C type                   | Array Element | Description (Reference)  | Unit (Format) | Allowed Range  |
|-----------------------|--------------------------|---------------|--|---------------|--|
| <code>model_id</code> | <code>xl_model_id</code> | -             | Model ID   | -             | -  |
| <code>mode</code>     | <code>long*</code>       | -             | Calculation mode selection   | -             | Select either:<br>· <code>XL_CALC_POS</code><br>· <code>XL_CALC_POS_VEL</code> |
| <code>pos[3]</code>   | <code>double</code>      | all           | Cartesian position vector (Earth fixed CS)   | m             | $r > a_{\text{ellipsoid}} - b_{\text{ellipsoid}}$                              |
| <code>vel[3]</code>   | <code>double</code>      | all           | Cartesian velocity vector (Earth fixed CS)<br>Dummy if <code>mode</code> is:<br>· <code>XL_CALC_POS</code> | m/s           | -  |

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

- Calculation mode selection: `mode`. See current document, section 6.2.

## 7.25.4 Output parameters

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

**Table 98: Output parameters of `xl_cart_to_geod` function**

| C name                       | C type   | Array Element | Description (Reference)  | Unit (Format) | Allowed Range            |
|------------------------------|----------|---------------|--|---------------|--------------------------|
| <code>xl_cart_to_geod</code> | long     | -             | Extended status flag   | -             | -                        |
| <code>lon</code>             | double * | -             | Geocentric longitude (Earth fixed CS)  | deg           | $\geq 0$<br>$< +360$     |
| <code>lat</code>             | double * | -             | Geodetic latitude (Earth fixed CS)   | deg           | $\geq -90$<br>$\leq +90$ |
| <code>h</code>               | double * | -             | Geodetic altitude (Earth fixed CS)   | m             | -                        |
| <code>lon_rate</code>        | double * | -             | Geocentric longitude rate (Earth fixed CS)<br>Returned only if <i>mode</i> is:<br>· <code>XL_CALC_POS_VEL</code> | deg/s         | -                        |
| <code>lat_rate</code>        | double * | -             | Geodetic latitude rate (Earth fixed CS)<br>Returned only if <i>mode</i> is:<br>· <code>XL_CALC_POS_VEL</code>    | deg/s         | -                        |
| <code>h_rate</code>          | double * | -             | Geodetic altitude rate (Earth fixed CS)<br>Returned only if <i>mode</i> is:<br>· <code>XL_CALC_POS_VEL</code>    | m/s           | -                        |

## 7.25.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_cart_to_geod` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_cart_to_geod` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 99: Error messages of `xl_cart_to_geod` function**

| Error type | Error message                                      | Cause and impact                                      | Error code                                   | Error No |
|------------|--|---|--|----------|
| ERR        | Internal computation error # 1                     | No calculation performed                              | <code>XL_CFI_CART_GEOD_FRAME_ERR</code>      | 0        |
| ERR        | Input vector out of valid range                    | No calculation performed                              | <code>XL_CFI_CART_GEOD_VECTOR_ERR</code>     | 1        |
| WARN       | Calculation mode ID is not correct                 | Calculation performed.<br>A message informs the user. | <code>XL_CFI_CART_GEOD_MODE_WARN</code>      | 2        |
| WARN       | Geocentric longitude set to 0 deg (ambiguous case) | Calculation performed.<br>A message informs the user. | <code>XL_CFI_CART_GEOD_AMBIGUITY_WARN</code> | 3        |
| WARN       | Internal computation warning #                     | Calculation performed.                                | <code>XL_CFI_CART_GEOD_AC</code>             | 4        |

|      |                                     |   |                                  |   |
|------|-------------------------------------|---|----------------------------------|---|
|      | 1                                   | A message informs the user.                           | CURACY_WARN                      |   |
| WARN | Internal computation warning #<br>2 | Calculation performed.<br>A message informs the user. | XL_CFI_CART_GEOD_ITERATIONS_WARN | 5 |
| WARN | Internal computation warning #<br>3 | Calculation performed.<br>A message informs the user. | XL_CFI_CART_GEOD_DE<br>FVAL_WARN | 6 |

## 7.26 xl\_kepl\_to\_cart

### 7.26.1 Overview

The xl\_kepl\_to\_cart CFI function transforms from keplerian to cartesian coordinates.

### 7.26.2 Calling interface

The calling interface of the xl\_kepl\_to\_cart CFI function is the following (input parameters are underlined>):

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long kepl_mode;
    double kepl_in[6];
    double pos_out[3], vel_out[3];
    long ierr[XL_NUM_ERR_KEPL_CART], status;

    status = xl_kepl_to_cart (&model_id, &kepl_mode,
                             kepl_in, pos_out,
                             vel_out, ierr);
}
```

### 7.26.3 Input parameters

The xl\_kepl\_to\_cart CFI function has the following input parameters:

*Table 100: Input parameters of xl\_kepl\_to\_cart function*

| C name     | C type       | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|------------|--------------|---------------|--|---------------|---------------|
| model_id   | xl_model_id* | -             | Model ID   | -             | Complete      |
| kepl_mode  | long*        | -             | Flag for selecting:<br>· Mean elements = XL_KEPLER_MEAN<br>· Osculating elements = XL_KEPLER_OSC | -             | Complete      |
| kepl_in[6] | double       | [0]           | Semi-major axis<br>(True of Date CS)   | m             | >= 0          |
|            |              | [1]           | Eccentricity<br>(True of Date CS)  | -             | [0,1)         |
|            |              | [2]           | Inclination<br>(True of Date CS)   | deg           | [0,180]       |

|  |     |  |     |         |
|--|-----|--|-----|---------|
|  | [3] | Right ascension of the ascending node<br>(True of Date CS) | deg | [0,360) |
|  | [4] | Argument of perigee<br>(True of Date CS)                   | deg | [0,360) |
|  | [5] | Mean anomaly<br>(True of Date CS)                          | deg | [0,360) |

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

- Kepler state vector model: `kepl_mode`. See section 6.2.

### 7.26.4 Output parameters

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

**Table 101: Output parameters of `xl_kepl_to_cart` function**

| C name                       | C type | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|------------------------------|--------|---------------|--|---------------|---------------|
| <code>xl_kepl_to_cart</code> | long   | -             | Status flag                                    | -             | -             |
| <code>pos_out[3]</code>      | double | all           | Cartesian position vector<br>(True of Date CS) | m             | -             |
| <code>vel_out[3]</code>      | double | all           | Cartesian velocity vector<br>(True of Date CS) | m/s           | -             |
| <code>ierr</code>            | long   | -             | Error vector                                   | -             | -             |

### 7.26.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_kepl_to_cart` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_kepl_to_cart` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 102: Error messages of `xl_kepl_to_cart` function**

| Error type | Error message  | Cause and impact         | Error code                  | Error No |
|------------|--|--------------------------|-----------------------------|----------|
| ERR        | Input semi-major axis $\leq 0$                               | No calculation performed | XL_CFI_K2C_A_ZERO_ERR       | 0        |
| ERR        | Input eccentricity $< 0$                                     | No calculation performed | XL_CFI_K2C_E_ZERO_ERR       | 1        |
| ERR        | Input eccentricity $> 1$                                     | No calculation performed | XL_CFI_K2C_E_ONE_ERR        | 2        |
| ERR        | Internal Error: Error in calling <code>XL_Mean_to_osc</code> | No calculation performed | XL_CFI_K2C_INTERNAL_M2O_ERR | 3        |
| ERR        | Internal computation error #1                                | No calculation performed | XL_CFI_K2C_COMPUTATI        | 4        |



|      |   |  |                              |   |
|------|---|--|------------------------------|---|
|      |   |  | ON_ERR                       |   |
| WARN | Internal Warning: Warning in calling XL_Mean_to_osc | Calculation performed. A message informs the user. | XL_CFI_K2C_INTERNAL_M2O_WARN | 5 |
| WARN | Kepler's equations not converged                    | Calculation performed. A message informs the user. | XL_CFI_K2C_NO_CONVERGED_WARN | 6 |

## 7.27 xl\_cart\_to\_kepl

### 7.27.1 Overview

The `xl_cart_to_kepl` CFI function transforms from cartesian to keplerian coordinates.

### 7.27.2 Calling interface

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

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long kepl_mode;
    double pos_in[3], vel_in[3];
    double kepl_out[6];
    long ierr[XL_NUM_ERR_CART_KEPL], status;

    status = xl_cart_to_kepl (&model_id,
                             pos_in, vel_in, &kepl_mode,
                             kepl_out, ierr);
}
```

### 7.27.3 Input parameters

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

*Table 103: Input parameters of xl\_cart\_to\_kepl function*

| C name    | C type      | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|-----------|-------------|---------------|--|---------------|---------------|
| model_id  | xl_model_id | -             | Model ID   | -             | -             |
| pos_in[3] | double      | all           | Cartesian position vector (True of Date CS)  | m             | -             |
| vel_in[3] | double      | all           | Cartesian velocity vector (True of Date CS)  | m/s           | -             |
| kepl_mode | long*       | -             | Flag for selecting:<br>· Mean elements = XL_KEPLER_MEAN<br>· Osculating elements = XL_KEPLER_OSC | -             | Complete      |

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

- Kepler state vector model: `kepl_mode`. See section 6.2.

### 7.27.4 Output parameters

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

**Table 104: Output parameters of `xl_cart_to_kepl` function**

| C name                       | C type | Array Element | Description (Reference)                                    | Unit (Format) | Allowed Range |
|------------------------------|--------|---------------|--|---------------|---------------|
| <code>xl_cart_to_kepl</code> | long   | -             | Status flag  | -             | -             |
| <code>kepl_out[6]</code>     | double | [0]           | Semi-major axis<br>(True of Date CS)                       | m             | $\geq 0$      |
|                              |        | [1]           | Eccentricity<br>(True of Date CS)                          | -             | [0,1)         |
|                              |        | [2]           | Inclination<br>(True of Date CS)                           | deg           | [0,180]       |
|                              |        | [3]           | Right ascension of the ascending node<br>(True of Date CS) | deg           | [0,360)       |
|                              |        | [4]           | Argument of perigee<br>(True of Date CS)                   | deg           | [0,360)       |
|                              |        | [5]           | Mean anomaly<br>(True of Date CS)                          | deg           | [0,360)       |
| <code>ierr</code>            | long   | -             | Error vector   | -             | -             |

### 7.27.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_cart_to_kepl` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_cart_to_kepl` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 105: Error messages of `xl_cart_to_kepl` function**

| Error type | Error message             | Cause and impact         | Error code                  | Error No |
|------------|---------------------------|--------------------------|-----------------------------|----------|
| ERR        | Earth's Mu < 0            | No calculation performed | XL_CFI_C2K_MU_ZERO_ERR      | 0        |
| ERR        | Input orbit radius = 0    | No calculation performed | XL_CFI_C2K_OR_ZERO_ERR      | 1        |
| ERR        | Input orbit velocity = 0  | No calculation performed | XL_CFI_C2K_OV_ZERO_ERR      | 2        |
| ERR        | Semi-major axis undefined | No calculation performed | XL_CFI_C2K_OA_UNDEFINED_ERR | 3        |

|      |   |  |                              |   |
|------|---|--|------------------------------|---|
| ERR  | Semi-major axis < 0                                 | No calculation performed                           | XL_CFI_C2K_OA_ZERO_ERR       | 4 |
| ERR  | Internal computation error #1                       | No calculation performed                           | XL_CFI_C2K_COMPUTATION_ERR   | 5 |
| ERR  | Internal Error: Error in calling XL_Osc_to_mean     | No calculation performed                           | XL_CFI_C2K_INTERNAL_O2M_ERR  | 6 |
| WARN | Inclination = 0 or 180 deg                          | Calculation performed. A message informs the user. | XL_CFI_C2K_OI_ZERO_WARN      | 7 |
| WARN | Eccentricity = 0                                    | Calculation performed. A message informs the user. | XL_CFI_C2K_OE_ZERO_WARN      | 8 |
| WARN | Internal Warning: Warning in calling XL_Osc_to_mean | Calculation performed. A message informs the user. | XL_CFI_C2K_INTERNAL_O2M_WARN | 9 |

## 7.28 xl\_cart\_to\_radec

### 7.28.1 Overview

The `xl_cart_to_radec` CFI function transforms cartesian coordinates to spherical coordinates:

- From equatorial cartesian coordinates to right ascension and declination.
- or
- From galactic cartesian coordinates to galactic longitude and latitude.

### 7.28.2 Calling interface

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

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long mode, cs_in;
    double pos[3], vel[3];
    double ra, dec, mu_ra, mu_dec, rad_vel, par;
    long ierr[XL_NUM_ERR_CART_RADEC], status;

    status = xl_cart_to_radec (&model_id, &mode, &cs_in, pos, vel,
                             &ra, &dec, &mu_ra, &mu_dec,
                             &rad_vel, &par, ierr);
}
```

### 7.28.3 Input parameters

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

**Table 106: Input parameters of `xl_cart_to_radec` function**

| C name   | C type      | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|----------|-------------|---------------|---|---------------|---------------|
| model_id | xl_model_id | -             | Model ID  | -             | -             |
| mode     | long*       | -             | Flag to select transformation, position or position and velocity: <ul style="list-style-type: none"> <li>• XL_CALC_POS</li> <li>• XL_CALC_POS_VEL</li> </ul> For galactic coordinates only position | -             | Complete      |

|        |        |     |  |     |                  |
|--------|--------|-----|--|-----|------------------|
|        |        |     | can be transformed.                              |     |                  |
| cs_in  | long*  | -   | Coordinate reference frame for the input vector. | -   | All except XL_EF |
| pos[3] | double | all | Cartesian position vector                        | m   | -                |
| vel[3] | double | all | Cartesian velocity vector                        | m/s | -                |

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

- Calculation mode: mode. See section 6.2
- Reference frame: cs\_in. See section 6.2

### 7.28.4 Output parameters

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

**Table 107: Output parameters of `xl_cart_to_radec` function**

| C name           | C type | Array Element | Description (Reference)                 | Unit (Format) | Allowed Range     |
|------------------|--------|---------------|---|---------------|-------------------|
| xl_cart_to_radec | long   | -             | Status flag                             | -             | -                 |
| ra               | double | -             | Right ascension (or galactic longitude) | rad           | [0, 2 $\pi$ )     |
| dec              | double | -             | Declination (or galactic latitude)      | rad           | $[-\pi/2, \pi/2]$ |
| mu_ra            | double | -             | Proper motion in the right ascension    | rad/century   | -                 |
| mu_dec           | double | -             | Proper motion in the declination        | rad/century   | -                 |
| rad_vel          | double | -             | Radial velocity                         | AU/century    | -                 |
| par              | double | -             | Parallax                                | rad           | [0, 2 $\pi$ )     |
| ierr             | long   | -             | Error vector                            | -             | -                 |

### 7.28.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_cart_to_radec` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_cart_to_radec` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 108: Error messages of `xl_cart_to_radec` function**

| Error type | Error message  | Cause and impact         | Error code                           | Error No |
|------------|--|--------------------------|--------------------------------------|----------|
| ERR        | Not possible to calculate velocity vector in galactic frame. | No calculation performed | XL_CFI_CART_TO_RADEC_WRONG_INPUT_ERR | 0        |
| ERR        | Mode input is not an allowed value.                          | No calculation performed | XL_CFI_CART_TO_RADEC_WRONG_MODE_ERR  | 1        |
| ERR        | cs_in input is not an allowed                                | No calculation performed | XL_CFI_CART_TO_RADEC                 | 2        |

---

|     |   |                          |  |   |
|-----|---|--------------------------|--|---|
|     | value.  |                          | WRONG_CS_IN_ERR                            |   |
| ERR | The frame's center is not an allowed position input | No calculation performed | XL_CFI_CART_TO_RADEC<br>WRONG_POSITION_ERR | 3 |

## 7.29 **xl\_radec\_to\_cart**

### 7.29.1 Overview

The `xl_radec_to_cart` CFI function transforms spherical coordinates to cartesian coordinates:

- From right ascension and declination to equatorial cartesian coordinates.

or

- From galactic longitude and latitude to galactic cartesian coordinates.

### 7.29.2 Calling interface

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

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long mode, cs_in;
    double pos[3], vel[3];
    double ra, dec, mu_ra, mu_dec, rad_vel, par;
    long ierr[XL_NUM_ERR_RADEC_CART], status;

    status = xl_radec_to_cart (&model_id, &mode,
                             &cs_in, &ra, &dec,
                             &mu_ra, &mu_dec,
                             &rad_vel, &par,
                             pos, vel, ierr);
}
```

### 7.29.3 Input parameters

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

**Table 109: Input parameters of `xl_radec_to_cart` function**

| C name   | C type      | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|----------|-------------|---------------|---|---------------|---------------|
| model_id | xl_model_id | -             | Model ID  | -             | -             |
| mode     | long*       | -             | Flag to select transformation, position or position and velocity: | -             | Complete      |



|         |        |   |   |             |                   |
|---------|--------|---|---|-------------|-------------------|
|         |        |   | <ul style="list-style-type: none"> <li>• XL_CALC_POS</li> <li>• XL_CALC_POS_VEL</li> </ul> For galactic coordinates only position can be transformed. |             |                   |
| cs_in   | long*  | - | Coordinate reference frame for the input vector.  | -           | All except XL_EF  |
| ra      | double | - | Right ascension (or galactic longitude)   | rad         | $[0, 2\pi)$       |
| dec     | double | - | Declination (or galactic latitude)  | rad         | $[-\pi/2, \pi/2]$ |
| mu_ra   | double | - | Proper motion in the right ascension  | rad/century | -                 |
| mu_dec  | double | - | Proper motion in the declination  | rad/century | -                 |
| rad_vel | double | - | Radial velocity   | AU/century  | -                 |
| par     | double | - | Parallax  | rad         | $[0, 2\pi)$       |

### 7.29.4 Output parameters

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

*Table 110: Output parameters of xl\_radec\_to\_cart function*

| C name           | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|------------------|--------|---------------|---------------------------|---------------|---------------|
| xl_radec_to_cart | long   | -             | Status flag               | -             | -             |
| pos[3]           | double | all           | Cartesian position vector | m             | -             |
| vel[3]           | double | all           | Cartesian velocity vector | m/s           | -             |
| ierr             | long   | -             | Error vector              | -             | -             |

### 7.29.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_radec_to_cart` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_radec_to_cart` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

*Table 111: Error messages of xl\_radec\_to\_cart function*

| Error type | Error message  | Cause and impact         | Error code                           | Error No |
|------------|--|--------------------------|--------------------------------------|----------|
| ERR        | Not possible to calculate velocity vector in galactic frame. | No calculation performed | XL_CFI_RADEC_TO_CART_WRONG_INPUT_ERR | 0        |
| ERR        | Mode input is not an allowed value.                          | No calculation performed | XL_CFI_RADEC_TO_CART_WRONG_MODE_ERR  | 1        |
| ERR        | cs_in input is not an allowed value."                        | No calculation performed | XL_CFI_RADEC_TO_CART_WRONG_CS_IN_ERR | 2        |
| ERR        | parallax can't be equal to zero.                             | No calculation performed | XL_CFI_RADEC_TO_CART_PAR_ERR         | 3        |



Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 154

---

## 7.30xl\_topocentric\_to\_ef

### 7.30.1Overview

The `xl_topocentric_to_ef` CFI function transforms topocentric azimuth and elevation to the Earth Fixed Reference frame.

### 7.30.2Calling interface

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

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long mode, deriv;
    double pos[3], vel[3];
    double azim, elev, range,
           azim_d, elev_d, range_d,
           ef_dir[3], ef_dir_d[3];
    long ierr[XL_NUM_ERR_TOP_TO_EF], status;

    status = xl_topocentric_to_ef(&model_id, &mode, &deriv, pos,
    vel,
                                &azim, &elev, &range,
                                &azim_d, &elev_d, &range_d,
                                ef_dir, ef_dir_d,
                                ierr);
}
```

### 7.30.3Input parameters

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

*Table 112: Input parameters of xl\_topocentric\_to\_ef function*

| C name   | C type      | Array Element | Description (Reference)  | Unit (Format) | Allowed Range   |
|----------|-------------|---------------|--|---------------|---|
| model_id | xl_model_id | -             | Model ID   | -             | -   |
| mode     | long        | -             | Flag to indicate if the input coordinates is location or a direction | -             | <ul style="list-style-type: none"> <li>XL_MODE_FLAG_LOCATION</li> </ul> |

|         |        |     |   |       |   |
|---------|--------|-----|---|-------|---|
|         |        |     |   |       | <ul style="list-style-type: none"> <li>• XL_MODE_FLAG_DIRECTION</li> </ul>          |
| • deriv | long   | -   | Flag to indicate if the 1st. derivative has to be computed. | -     | <ul style="list-style-type: none"> <li>• XL_NO_DER</li> <li>• XL_DER_1ST</li> </ul> |
| • pos   | double | all | Position of the topocentric CS in the EF CS                 | m     | -   |
| vel     | double | all | Velocity of the topocentric CS in the EF CS                 | m/s   | -   |
| azim    | double | -   | Azimuth   | deg   | [0, 360)  |
| elev    | double | -   | Elevation   | deg   | [-90, +90]  |
| range   | double | -   | Distance  | m     | -   |
| azim_d  | double | -   | Azimuth rate  | deg/s | [0, 360)  |
| elev_d  | double | -   | Elevation rate  | deg/s | [-90, +90]  |
| range_d | double | -   | Range rate  | m/s   | -   |

### 7.30.4 Output parameters

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

**Table 113: Output parameters of `xl_topocentric_to_ef` function**

| C name                            | C type | Array Element | Description (Reference)         | Unit (Format) | Allowed Range |
|-----------------------------------|--------|---------------|---------------------------------|---------------|---------------|
| <code>xl_topocentric_to_ef</code> | long   | -             | Status flag                     | -             | -             |
| <code>ef_dir</code>               | double | all           | Cartesian position vector in EF | m             | -             |
| <code>ef_dir_d</code>             | double | all           | Cartesian velocity vector in EF | m/s           | -             |
| <code>ierr</code>                 | long   | -             | Error vector                    | -             | -             |

### 7.30.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_topocentric_to_ef` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_topocentric_to_ef` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 114: Error messages of `xl_topocentric_to_ef` function**

| Error type | Error message                              | Cause and impact         | Error code                            | Error No |
|------------|--|--------------------------|---------------------------------------|----------|
| ERR        | Wrong for the parameter Location/Direction | No calculation performed | XL_CFI_TOP_TO_EF_WRONG_MODE_FLAG_ERR  | 0        |
| ERR        | Wrong parameter for the derivative         | No calculation performed | XL_CFI_TOP_TO_EF_WRONG_DERIV_FLAG_ERR | 1        |

---

|     |   |                          |                                   |   |
|-----|---|--------------------------|-----------------------------------|---|
| ERR | Could not convert input vector for the topocentric center to geodetic coordinates | No calculation performed | XL_CFI_TOP_TO_EF_CART_TO_GEOD_ERR | 2 |
| ERR | Could not get the pointing direction from the input Azimuth and elevation         | No calculation performed | XL_CFI_TOP_TO_EF_POINTING_DIR_ERR | 3 |

## 7.31xl\_ef\_to\_topocentric

### 7.31.1Overview

The `xl_ef_to_topocentric` CFI function transforms Earth Fixed coordinates to topocentric coordinates for a given ground position.

### 7.31.2Calling interface

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

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long mode, deriv;
    double pos[3], vel[3];
    double azim, elev, range,
           azim_d, elev_d, range_d,
           ef_dir[3], ef_dir_d[3];
    long ierr[XL_NUM_ERR_TOP_TO_EF], status;

    status = xl_ef_to_topocentric(&model_id,
                                &mode, &deriv, pos, vel,
                                ef_dir, ef_dir_d,
                                &azim, &elev, &range,
                                &azim_d, &elev_d, &range_d,
                                ierr);
}
```

### 7.31.3Input parameters

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

*Table 115: Input parameters of xl\_ef\_to\_topocentric function*

| C name   | C type      | Array Element | Description (Reference)  | Unit (Format) | Allowed Range   |
|----------|-------------|---------------|--|---------------|---|
| model_id | xl_model_id | -             | Model ID   | -             | -   |
| mode     | long        | -             | Flag to indicate if the input coordinates is location or a direction | -             | <ul style="list-style-type: none"> <li>XL_MODE_FLAG_LOCATION</li> </ul> |

|          |        |     |   |     |   |
|----------|--------|-----|---|-----|---|
|          |        |     |   |     | <ul style="list-style-type: none"> <li>XL_MODE_FLAG_DIRECTION</li> </ul>        |
| • deriv  | long   | -   | Flag to indicate if the 1st. derivative has to be computed. |     | <ul style="list-style-type: none"> <li>XL_NO_DER</li> <li>XL_DER_1ST</li> </ul> |
| • pos    | double | all | Position of the topocentric CS in the EF CS                 | m   |   |
| vel      | double | all | Velocity of the topocentric CS in the EF CS                 | m/s |   |
| ef_dir   | double | all | Cartesian position vector in EF                             | m   |   |
| ef_dir_d | double | all | Cartesian velocity vector in EF                             | m/s |   |

### 7.31.4 Output parameters

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

*Table 116: Output parameters of xl\_ef\_to\_topocentric function*

| C name               | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|----------------------|--------|---------------|-------------------------|---------------|---------------|
| xl_topocentric_to_ef | long   | -             | Status flag             | -             | -             |
| azim                 | double | -             | Azimuth                 | deg           | [0, 360]      |
| elev                 | double | -             | Elevation               | deg           | [-90, +90]    |
| range                | double | -             | Distance                | m             | -             |
| azim_d               | double | -             | Azimuth rate            | deg/s         | [0, 360]      |
| elev_d               | double | -             | Elevation rate          | deg/s         | [-90, +90]    |
| range_d              | double | -             | Range rate              | m/s           | -             |
| ierr                 | long   | -             | Error vector            | -             | -             |

### 7.31.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_ef_to_topocentric` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 `xl_ef_to_topocentric` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

*Table 117: Error messages of xl\_ef\_to\_topocentric function*

| Error type | Error message  | Cause and impact         | Error code                           | Error No |
|------------|--|--------------------------|--------------------------------------|----------|
| ERR        | Wrong for the parameter Location/Direction                   | No calculation performed | XL_CFI_EF_TO_TOP_WRONG_MODE_FLAG_ERR | 0        |
| ERR        | Could not convert input vector for the topocentric center to | No calculation performed | XL_CFI_EF_TO_TOP_CART_TO_GEOD_ERR    | 1        |

---

|     |  |                          |                                       |   |
|-----|--|--------------------------|---------------------------------------|---|
|     | geodetic coordinates                       |                          |                                       |   |
| ERR | Wrong parameter for the derivative         | No calculation performed | XL_CFI_EF_TO_TOP_WRONG_DERIV_FLAG_ERR | 2 |
| ERR | Error when computing Azimuth and Elevation | No calculation performed | XL_CFI_EF_TO_TOP_DIR_POINTING_ERR     | 3 |



## 7.32xl\_sun

### 7.32.1Overview

The `xl_sun` CFI function calculates the position and velocity vector of the Sun in the Earth Fixed coordinate system.

### 7.32.2Calling interface

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

```
#include <explorer_lib.h>
{
    long time_ref;
    double time, sun_pos[3], sun_vel[3];
    xl_model_id model_id = {NULL};
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_SUN], status;

    status = xl_sun(&model_id,
                   &time_id, &time_ref, &time,
                   sun_pos, sun_vel,
                   ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_sun_run(&run_id, &time_ref, &time, sun_pos, sun_vel,
                       ierr);
}
```

### 7.32.3Input parameters

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

**Table 118: Input parameters of xl\_sun function**

| C name   | C type       | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|----------|--------------|---------------|-------------------------|---------------|---------------|
| model_id | xl_model_id* | -             | Model ID                | -             | -             |

|          |             |   |  |                                  |                          |
|----------|-------------|---|--|----------------------------------|--------------------------|
| time_id  | xl_time_id* | - | Structure that contains the time correlations. | -                                | -                        |
| time_ref | long *      | - | Initial time reference ID                      | -                                | Any except XL_TIME_UNDEF |
| time     | double*     | - | Input time                                     | Decimal days (Processing format) | [-18262.0,36524.0]       |

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

- Time reference ID: time\_ref. See [GEN\_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

### 7.32.4 Output parameters

The output parameters of the **xl\_sun** CFI function are:

**Table 119: Output parameters of xl\_sun function**

| C name     | C type | Array Element | Description (Reference)                          | Unit (Format) | Allowed Range |
|------------|--------|---------------|--|---------------|---------------|
| xl_sun     | long   | -             | Status flag                                      | -             | -             |
| sun_pos[3] | double | all           | Position vector of the Sun in the Earth Fixed CS | m             | -             |
| sun_vel[3] | double | all           | Velocity vector of the Sun in the Earth Fixed CS | m/s           | -             |
| ierr       | long   | -             | Error vector                                     | -             | -             |

### 7.32.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_sun** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 **xl\_sun** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 120: Error messages of xl\_sun function**

| Error type | Error message                          | Cause and impact         | Error code              | Error No |
|------------|--|--------------------------|-------------------------|----------|
| ERR        | Input time reference ID is not correct | No calculation performed | XL_CFI_SUN_REF_ERR      | 0        |
| ERR        | Input date is out of range             | No calculation performed | XL_CFI_SUN_DAY_ERR      | 1        |
| ERR        | Time Reference not initialised         | No calculation performed | XL_CFI_SUN_REF_INIT_ERR | 2        |
| ERR        | Error in calling                       | No calculation performed | XL_CFI_SUN_SUN_ERR      | 3        |



Code: EO-MA-DMS-GS-0003

Date: 06/02/12

Issue: 4.3

Page: 163

---

|               |  |  |  |
|---------------|--|--|--|
| XL_Sun_PosVel |  |  |  |
|---------------|--|--|--|

## 7.33xl\_moon

### 7.33.1Overview

The `xl_moon` CFI function calculates the position and velocity vector of the Moon in the Earth Fixed coordinate system.

### 7.33.2Calling interface

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

```
#include <explorer_lib.h>
{
    long time_ref;
    double time, moon_pos[3], moon_vel[3];
    xl_model_id model_id = {NULL};
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_MOON], status;

    status = xl_moon(&model_id,
                    &time_id, &time_ref, &time,
                    moon_pos, moon_vel,
                    ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_moon_run(&run_id, &time_ref, &time,
                        moon_pos, moon_vel,
                        ierr);
}
```

### 7.33.3Input parameters

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

**Table 121: Input parameters of xl\_moon function**

| C name   | C type       | Array Element | Description (Reference)                        | Unit (Format)                    | Allowed Range            |
|----------|--------------|---------------|--|----------------------------------|--------------------------|
| model_id | xl_model_id* | -             | Model ID                                       | -                                | -                        |
| time_id  | xl_time_id*  | -             | Structure that contains the time correlations. | -                                | -                        |
| time_ref | long *       | -             | Initial time reference ID                      | -                                | Any except XL_TIME_UNDEF |
| time     | double*      | -             | Input time                                     | Decimal days (Processing format) | [-18262.0,36524.0]       |

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

- Time reference ID: time\_ref. See [GEN\_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details).

### 7.33.4 Output parameters

The output parameters of the **xl\_moon** CFI function are:

**Table 122: Output parameters of xl\_moon function**

| C name      | C type | Array Element | Description (Reference)                           | Unit (Format) | Allowed Range |
|-------------|--------|---------------|---|---------------|---------------|
| xl_moon     | long   | -             | Status flag                                       | -             | -             |
| moon_pos[3] | double | all           | Position vector of the Moon in the Earth Fixed CS | m             | -             |
| moon_vel[3] | double | all           | Velocity vector of the Moon in the Earth Fixed CS | m/s           | -             |
| ierr        | long   | -             | Error vector                                      | -             | -             |

### 7.33.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_moon** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 **xl\_moon** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 123: Error messages of xl\_moon function**

| Error | Error message | Cause and impact | Error code | Error |
|-------|---------------|------------------|------------|-------|
|-------|---------------|------------------|------------|-------|

| <b>type</b> |  |                          |                          | <b>No</b> |
|-------------|--|--------------------------|--------------------------|-----------|
| ERR         | Input time reference ID is not correct | No calculation performed | XL_CFI_MOON_REF_ERR      | 0         |
| ERR         | Input date is out of range             | No calculation performed | XL_CFI_MOON_DAY_ERR      | 1         |
| ERR         | Time Reference not initialised         | No calculation performed | XL_CFI_MOON_REF_INIT_ERR | 2         |
| ERR         | Error in calling XL_Moon_PosVel        | No calculation performed | XL_CFI_MOON_MOON_ERR     | 3         |

## 7.34xl\_planet

### 7.34.1Overview

The **xl\_planet** CFI function calculates the position and velocity vector of a planet in the Earth Fixed coordinate system.

### 7.34.2Calling interface

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

```
#include <explorer_lib.h>
{
    long sat_id, planet, time_ref;
    double time, planet_pos[3], planet_vel[3];
    xl_model_id model_id = {NULL};
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_PLANET], status;

    status =    xl_planet(&model_id, &time_id,
                        &planet, &time_ref, &time,
                        planet_pos, planet_vel, ierr);

    /* Or, using the run_id */
    long run_id;

    status =    xl_planet_run(&run_id, &planet, &time_ref, &time,
                            planet_pos, planet_vel, ierr);
}
```

### 7.34.3Input parameters

The **xl\_planet** CFI function has the following input parameters:

**Table 124: Input parameters of xl\_planet function**

| C name   | C type       | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|----------|--------------|---------------|-------------------------|---------------|---------------|
| model_id | xl_model_id* | -             | Model ID                | -             | -             |
| time_id  | xl_time_id*  | -             | Structure that contains | -             | -             |

|          |         |   |                           |                                     |                             |
|----------|---------|---|---------------------------|-------------------------------------|-----------------------------|
|          |         |   | the time correlations.    |                                     |                             |
| planet   | long *  | - | Planet ID                 | -                                   | Complete                    |
| time_ref | long *  | - | Initial time reference ID | -                                   | Any except<br>XL_TIME_UNDEF |
| time     | double* | - | Input time                | Decimal days<br>(Processing format) | [-18262.0,36524.0]          |

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

- Time reference ID: time\_ref. See [GEN\_SUM].
- Planet ID: planet. Current document, section 6.2.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

### 7.34.4 Output parameters

The output parameters of the **xl\_planet** CFI function are:

*Table 125: Output parameters of xl\_planet function*

| C name        | C type | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|---------------|--------|---------------|--|---------------|---------------|
| xl_planet     | long   | -             | Status flag  | -             | -             |
| planet_pos[3] | double | all           | Position vector of the Planet in the Earth Fixed coordinate system | m             | -             |
| planet_vel[3] | double | all           | Velocity vector of the Planet in the Earth Fixed coordinate system | m/s           | -             |
| ierr          | long   | -             | Error vector   | -             | -             |

### 7.34.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_planet** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 **xl\_planet** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

*Table 126: Error messages of xl\_planet function*

| Error type | Error message                          | Cause and impact         | Error code            | Error No |
|------------|--|--------------------------|-----------------------|----------|
| ERR        | Input time reference ID is not correct | No calculation performed | XL_CFI_PLANET_REF_ERR | 0        |



|      |  |   |                                |   |
|------|--|---|--------------------------------|---|
| ERR  | Input date is out of range                               | No calculation performed                              | XL_CFI_PLANET_DAY_ER<br>R      | 1 |
| ERR  | Time Reference not initialised                           | No calculation performed                              | XL_CFI_PLANET_REF_INI<br>T_ERR | 2 |
| ERR  | Planet code is not correct                               | No calculation performed                              | XL_CFI_PLANET_PLANET<br>ERR    | 3 |
| WARN | Internal Warning: XL_Planets<br>solution didn't converge | Calculation performed.<br>A message informs the user. | XL_CFI_PLANET_CONV_<br>WARN    | 4 |

## 7.35 xl\_star\_radec

### 7.35.1 Overview

The `xl_star_radec` CFI function calculates the right ascension and declination of a star in the True of Date coordinate system.

### 7.35.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long time_ref;
    double time, ra0, dec0, mu_ra, mu_dec;
    double rad_vel, par, ra, dec;
    xl_time_id time_id = {NULL};
    xl_model_id model_id = {NULL};
    long ierr[XL_NUM_ERR_STAR], status;

    status = xl_star_radec(&model_id, &time_id,
                          &time_ref, &time, &ra0, &dec0,
                          &mu_ra, &mu_dec, &rad_vel, &par,
                          &ra, &dec, ierr);

    /* Or, using the run_id */
    long run_id;

    status = xl_star_radec_run(&run_id, &time_ref, &time, &ra0, &dec0,
                              &mu_ra, &mu_dec, &rad_vel, &par,
                              &ra, &dec, ierr);
}
```

### 7.35.3 Input parameters

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

**Table 127: Input parameters of xl\_star\_radec function**

| C name   | C type       | Array Element | Description (Reference)   | Unit (Format)                    | Allowed Range            |
|----------|--------------|---------------|---|----------------------------------|--------------------------|
| model_id | xl_model_id* | -             | Model ID  | -                                | -                        |
| time_id  | xl_time_id*  | -             | Structure that contains the time correlations.                                  | -                                | -                        |
| time_ref | long *       | -             | Initial time reference ID   | -                                | Any except XL_TIME_UNDEF |
| time     | double*      | -             | Input time  | Decimal days (Processing format) | [-18262.0,36524.0]       |
| ra0      | double *     | -             | Right ascension of the star at J2000.0 (Barycentric Mean of 2000.0 CS)          | rad                              | [0,2 $\pi$ )             |
| dec0     | double *     | -             | Declination of the star at J2000.0 (Barycentric Mean of 2000.0 CS)              | rad                              | [- $\pi/2$ , $\pi/2$ ]   |
| mu_ra    | double *     | -             | Proper motion in the right ascension at J2000.0 (Barycentric Mean of 2000.0 CS) | rad/century                      | -                        |
| mu_dec   | double *     | -             | Proper motion in the declination at J2000.0 (Barycentric Mean of 2000.0 CS)     | rad/century                      | -                        |
| rad_vel  | double *     | -             | Radial velocity of the star at J2000.0 (Barycentric Mean of 2000.0 CS)          | AU/century                       | -                        |
| par      | double *     | -             | Parallax of the star at J2000.0 (Barycentric Mean of 2000.0 CS)                 | rad                              | -                        |

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

- Time reference ID: time\_ref. See [GEN\_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

### 7.35.4 Output parameters

The output parameters of the xl\_star\_radec CFI function are:

**Table 128: Output parameters of xl\_star\_radec function**

| C name        | C type   | Array Element | Description (Reference)   | Unit (Format) | Allowed Range          |
|---------------|----------|---------------|---|---------------|------------------------|
| xl_star_radec | long     | -             | Status flag   | -             | -                      |
| ra            | double * | -             | Right ascension of the star at specified time (True of Date CS) | rad           | [0,2 $\pi$ )           |
| dec           | double * | -             | Declination of the star at specified time (True of Date CS)     | rad           | [- $\pi/2$ , $\pi/2$ ] |
| ierr          | long     | -             | Error vector  | -             | -                      |

### 7.35.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_star\_radec** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 **xl\_star\_radec** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 129: Error messages of xl\_star\_radec function**

| Error type | Error message                          | Cause and impact                                   | Error code                      | Error No |
|------------|--|--|---------------------------------|----------|
| ERR        | Input time reference ID is not correct | No calculation performed                           | XL_CFI_STAR_RADEC_REF_ERR       | 0        |
| ERR        | Input date is out of range             | No calculation performed                           | XL_CFI_STAR_RADEC_DAY_ERR       | 1        |
| ERR        | Time Reference not initialised         | No calculation performed                           | XL_CFI_STAR_RADEC_REF_INIT_ERR  | 2        |
| ERR        | Error in calling XL_Star               | No calculation performed                           | XL_CFI_STAR_RADEC_STAR_ERR      | 3        |
| ERR        | Error in calling XL_Dir_Pointing       | No calculation performed                           | XL_CFI_STAR_RADEC_DIRPOINT_ERR  | 4        |
| WARN       | Warning in calling XL_Dir_Pointing     | Calculation performed. A message informs the user. | XL_CFI_STAR_RADEC_DIRPOINT_WARN | 5        |

The declination is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see section 10).

## 7.36 *xl\_star\_catalog*

### 7.36.1 Overview

The *xl\_star\_catalog* CFI function calculates the right ascension and declination of a star in a selected star catalogue.

### 7.36.2 Calling interface

The calling interface of the *xl\_star\_catalog* CFI function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long time_ref, catalog_in, cs_out, mode;
    double time, ra0, dec0, mu_ra0, mu_dec0;
    double rad_vel0, par0, ra, dec;
    xl_model_id model_id = {NULL};
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_STAR_CATALOG], status;

    status = xl_star_catalog(&model_id, &time_id,
                           &time_ref, &time, &mode,
                           &catalog_in, &catalog_out, &ra0, &dec0,
                           &mu_ra0, &mu_dec0, &rad_vel0, &par0,
                           &ra, &dec, ierr);

    /* Or, using the run_id */
    long run_id;
    status = xl_star_catalog_run(&run_id, &time_ref, &time, &mode,
                                &catalog_in, &catalog_out, &ra0, &dec0,
                                &mu_ra0, &mu_dec0, &rad_vel0, &par0,
                                &ra, &dec, ierr);
}
```

### 7.36.3 Input parameters

The *xl\_star\_catalog* CFI function has the following input parameters:

*Table 130: Input parameters of xl\_star\_catalog function*

| C name | C type | Array | Description | Unit | Allowed Range |
|--------|--------|-------|-------------|------|---------------|
|--------|--------|-------|-------------|------|---------------|

|             |              | Element | (Reference)   | (Format)                         |                          |
|-------------|--------------|---------|---|----------------------------------|--------------------------|
| model_id    | xl_model_id* | -       | Model ID.   | -                                | -                        |
| time_id     | xl_time_id*  | -       | Structure that contains the time correlations.            | -                                | -                        |
| time_ref    | long *       | -       | Initial time reference ID                                 | -                                | Any except XL_TIME_UNDEF |
| time        | double*      | -       | Input time  | Decimal days (Processing format) | [-18262.0,36524.0]       |
| mode        | long*        | -       |   |                                  |                          |
| catalog_in  | long*        | -       | Input star catalog  | -                                | All                      |
| catalog_out | long*        | -       | Output coordinate frame                                   | -                                | All                      |
| ra0         | double *     | -       | Right ascension of the star in the input catalog          | rad                              | [0,2 $\pi$ )             |
| dec0        | double *     | -       | Declination of the star in the input catalog              | rad                              | $[-\pi/2, \pi/2]$        |
| mu_ra0      | double *     | -       | Proper motion in the right ascension in the input catalog | rad/century                      | -                        |
| mu_dec0     | double *     | -       | Proper motion in the declination in the input catalog     | rad/century                      | -                        |
| rad_vel0    | double *     | -       | Radial velocity of the star in the input catalog          | AU/century                       | -                        |
| par0        | double *     | -       | Parallax of the star in the input catalog                 | rad                              | -                        |

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

- Time reference ID: time\_ref. See [GEN\_SUM].
- Star catalog ID: catalog\_in. See section 6.2
- Reference frame: cs\_out. See section 6.2

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

### 7.36.4 Output parameters

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

**Table 131: Output parameters of xl\_star\_catalog function**

| C name          | C type   | Array Element | Description (Reference)  | Unit (Format) | Allowed Range     |
|-----------------|----------|---------------|--|---------------|-------------------|
| xl_star_catalog | long     | -             | Status flag  | -             | -                 |
| ra              | double * | -             | Right ascension of the star at specified time in the out_cs reference frame. | rad           | [0,2 $\pi$ )      |
| dec             | double * | -             | Declination of the star at specified time in the out_cs reference frame.     | rad           | $[-\pi/2, \pi/2]$ |
| ierr            | long     | -             | Error vector   | -             | -                 |

### 7.36.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_star\_catalog** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 **xl\_star\_catalog** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 132: Error messages of xl\_star\_catalog function**

| Error type | Error message                         | Cause and impact         | Error code                                   | Error No |
|------------|---------------------------------------|--------------------------|--|----------|
| ERR        | Wrong input catalog                   | No calculation performed | XL_CFI_STAR_CATALOG_WRONG_INPUT_CATALOG_ERR  | 0        |
| ERR        | Wrong output catalog                  | No calculation performed | XL_CFI_STAR_CATALOG_WRONG_OUTPUT_CATALOG_ERR | 1        |
| ERR        | Error in xl_star_radec                | No calculation performed | XL_CFI_STAR_CATALOG_STAR_RADEC_ERR           | 2        |
| ERR        | Error when converting from FK4 to FK5 | No calculation performed | XL_CFI_STAR_CATALOG_FK4_TO_FK5_ERR           | 3        |
| ERR        | Error in xl_radec_to_cart             | No calculation performed | XL_CFI_STAR_CATALOG_RADEC_TO_CART_ERR        | 4        |
| ERR        | Error in xl_change_coordinate_cs      | No calculation performed | XL_CFI_STAR_CATALOG_CHANGE_CART_CS_ERR       | 5        |
| ERR        | Error in xl_cart_to_radec             | No calculation performed | XL_CFI_STAR_CATALOG_CART_TO_RADEC_ERR        | 6        |

## 7.37xl\_geod\_distance

### 7.37.1Overview

The `xl_geod_distance` CFI function calculates the geodesic distance between two points that lay on the same ellipsoid, and the azimuth of the related geodesic line at both points. See diagram below.

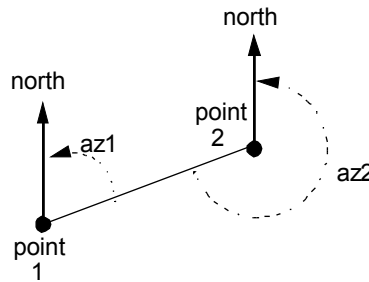


Figure 3: Azimuth figures returned by `xl_geod_distance` function

### 7.37.2Calling interface

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

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    double lon1, lat1, lon2, lat2, h;
    double distance, az_1_to_2, az_2_to_1;
    long status;
    status = xl_geod_distance (&model_id,
                               &lon1, &lat1, &lon2, &lat2, &h,
                               &distance, &az_1_to_2,
                               &az_2_to_1);
}
```

### 7.37.3Input parameters

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

Table 133: Input parameters of `xl_geod_distance` function

| C name   | C type      | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|----------|-------------|---------------|-------------------------|---------------|---------------|
| model_id | xl_model_id | -             | Model ID                | -             | -             |



|      |          |   |   |     |   |
|------|----------|---|---|-----|---|
| lon1 | double * | - | Geocentric longitude of the first input point (Earth fixed CS)  | deg | [0,360)                                       |
| lat1 | double * | - | Geodetic latitude of the first input point (Earth fixed CS)     | deg | [-90,90]                                      |
| lon2 | double * | - | Geocentric longitude of the second input point (Earth fixed CS) | deg | [0,360)                                       |
| lat2 | double * | - | Geodetic latitude of the second input point (Earth fixed CS)    | deg | [-90,90]                                      |
| h    | double * | - | Geodetic altitude of both input points (Earth fixed CS)         | m   | $h \geq -b_{WGS}$<br>(satellite ID dependent) |

### 7.37.4 Output parameters

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

**Table 134: Output parameters of `xl_geod_distance` function**

| C name                        | C type   | Array Element | Description (Reference)  | Unit (Format) | Allowed Range       |
|-------------------------------|----------|---------------|--|---------------|---------------------|
| <code>xl_geod_distance</code> | long     | -             | Extended status flag   | -             | -                   |
| <code>distance</code>         | double * | -             | Geodesic distance between the two input points (Earth fixed CS)  | m             | $\geq 0$            |
| <code>az_1_to_2</code>        | double * | -             | Azimuth of the geodesic line from point 1 to point 2 (Topocentric CS)  | deg           | [0,360)             |
| <code>az_2_to_1</code>        | double * | -             | Azimuth of the geodesic line from point 2 to point 1 (Topocentric CS)<br>Note that $az_2 = az_1 + 180$ approximately | deg           | $\geq 0$<br>$< 360$ |

### 7.37.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_geod_distance` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_geod_distance` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 135: Error messages of xl\_geod\_distance function**

| Error type | Error message   | Cause and impact                                   | Error code                             | Error No |
|------------|---|--|--|----------|
| ERR        | Different altitudes in the two points   | No calculation performed                           | XL_CFI_GEOD_DIST_ALTITUDE_ERR          | 0        |
| ERR        | Calculation not performed in XL_Geo_Car   | No calculation performed                           | XL_CFI_GEOD_DIST_GEO_CAR_ERR           | 1        |
| ERR        | Calculation not performed in XL_Pt_Dir_Range  | No calculation performed                           | XL_CFI_GEOD_DIST_DIR_RANGE_ERR         | 2        |
| ERR        | No solution returned by XL_Dir_Pointing   | No calculation performed                           | XL_CFI_GEOD_DIST_DIR_POINTING_ERR      | 3        |
| WARN       | Antipodal points. Two possible azimuth values (0 or 180). Selected value is 0.0 deg | Calculation performed. A message informs the user. | XL_CFI_GEOD_DIST_ANTIPODAL_POINTS_WARN | 4        |
| WARN       | Default values returned by XL_Dir_Pointing  | Calculation performed. A message informs the user. | XL_CFI_GEOD_DIST_DIR_POINTING_WARN     | 5        |

The altitude of the two points is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see section 10).

For antipodal points, a little variation of the input coordinates may lead to incoherent values for the output distance, depending on the point location on the ellipsoid.

## 7.38 xl\_time\_get\_leap\_second\_info

### 7.38.1 Overview

The `xl_time_get_leap_second_info` CFI function retrieves the leap second location (if any) in the initialised time range.

In order to avoid ambiguities the instant of Leap Second insertion is given both as the instant just before insertion (i.e. when the LS start) and the instant just after insertion (i.e. when the LS ends).

As an example, in the case of the (positive) LS inserted on January 1st, 1999, the function would return (if `ascii_id_out = XL_ASCII_STD_REF_MICROSEC`):

```

  leap_flag = 1
  ascii_utc_time_before_leap = UTC=1998-12-31_23:59:60.000000
  ascii_utc_time_after_leap = UTC=1999-01-01_00:00:00.000000

```

In the case of a negative LS, inserted as an example on January 1st, 2009, the function would return (if `ascii_id_out = XL_ASCII_STD_REF_MICROSEC`):

```

  leap_flag = -1
  ascii_utc_time_before_leap = UTC=2008-12-31_23:59:58.000000
  ascii_utc_time_after_leap = UTC=2009-01-01_00:00:00.000000

```

Note that, if the time correlations were initialised with an Orbit Scenario File, LS could be wrongly calculated (see section 7.1.).

### 7.38.2 Calling interface

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

```

#include <explorer_lib.h>

{
    long ascii_id_out, leap_flag;
    char ascii_utc_time_before_leap[XL_TIME_ASCII_DIM_MAX];
    char ascii_utc_time_after_leap[XL_TIME_ASCII_DIM_MAX]
    xl_time_id time_id = {NULL};
    long ierr[XL_NUM_ERR_LEAP_INFO], status;

    status = xl_time_get_leap_second_info(&time_id, &ascii_id_out,
                                         &leap_flag, ascii_utc_time_before_leap,
                                         ascii_utc_time_after_leap, ierr);

    /* Or, using the run_id */
    long run_id;
    status = xl_time_get_leap_second_info_run(&run_id, &ascii_id_out,

```

```

    &leap_flag,      ascii_utc_time_before_leap,
    ascii_utc_time_after_leap, ierr);
}

```

The `XL_TIME_ASCII_DIM_MAX` and `XL_NUM_ERR_LEAP_INFO` constants are defined in the file `explorer_lib.h`.

### 7.38.3 Input parameters

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

**Table 136: Input parameters of `xl_time_get_leap_second_info` function**

| C name                    | C type                   | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|---------------------------|--------------------------|---------------|--|---------------|---------------|
| <code>time_id</code>      | <code>xl_time_id*</code> | -             | Structure that contains the time correlations. | -             | -             |
| <code>ascii_id_out</code> | <code>long *</code>      | -             | ASCII format ID for output                     | -             | Complete      |

It is possible to use enumeration values rather than integer values for the input argument:

- ASCII format ID: `ascii_id_out`. Current document, section 6.2.

### 7.38.4 Output parameters

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

**Table 137: Output parameters of `xl_time_get_leap_second_info` function**

| C name                                    | C type              | Array Element           | Description (Reference)   | Unit (Format)           | Allowed Range   |
|---|---------------------|-------------------------|---|-------------------------|---|
| <code>xl_time_get_leap_second_info</code> | <code>long</code>   | -                       | Status flag   | -                       | -   |
| <code>leap_flag</code>                    | <code>long *</code> | -                       | Flag for leap second presence within time initialization data                   | -                       | -1 = Negative Leap Second (a LS has been removed) (very rare case)<br>0 = No leap second within initialization data<br>+1 = Positive Leap Second (a LS has been added) (usual case) |
| <code>ascii_utc_time_before_leap</code>   | <code>char</code>   | See Table 4 and Table 5 | UTC time just before leap second insertion (dummy if <code>leap_flag=0</code> ) | See Table 4 and Table 5 | See Table 4 and Table 5   |
| <code>ascii_utc_time_after_leap</code>    | <code>char</code>   | See Table 4 and Table 5 | UTC time just after leap second insertion (dummy if <code>leap_flag=0</code> )  | See Table 4 and Table 5 | See Table 4 and Table 5   |
| <code>ierr</code>                         | <code>long</code>   | -                       | Error vector  | -                       | -   |

Note that if more than one leap second is contained within the time initialization data for the selected satellite, only the last (most recent) one is returned.

No more than one leap second is likely to be found in the data, unless the range of time initialization span more than one year (a total of 23 leap seconds have been inserted until 2002, since the system was introduced in 1972).

### 7.38.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_time_get_leap_second_info` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_time_get_leap_second_info` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 138: Error messages of `xl_time_get_leap_second_info` function**

| Error type | Error message  | Cause and impact  | Error code                                     | Error No |
|------------|--|---|--|----------|
| ERR        | Output ascii format ID is not correct                | No calculation performed                                | XL_CFI_TIME_LEAP_SEC<br>OND_ASCII_OUT_ERR      | 0        |
| ERR        | Satellite ID and output format ID are not compatible | No calculation performed                                | XL_CFI_TIME_LEAP_SEC<br>OND_COMP_OUT_ERR       | 1        |
| ERR        | Error in adding times in Processing format           | No calculation performed                                | XL_CFI_TIME_LEAP_SEC<br>OND_ADD_ERR            | 2        |
| ERR        | Error in converting from Processing to ASCII format  | No calculation performed                                | XL_CFI_TIME_LEAP_SEC<br>OND_P2A_ERR            | 3        |
| WARN       | Time Reference not initialised                       | No calculation performed<br>A message informs the user. | XL_CFI_TIME_LEAP_SEC<br>OND_TIME_REF_INIT_WARN | 4        |

## 7.39 xl\_euler\_to\_matrix

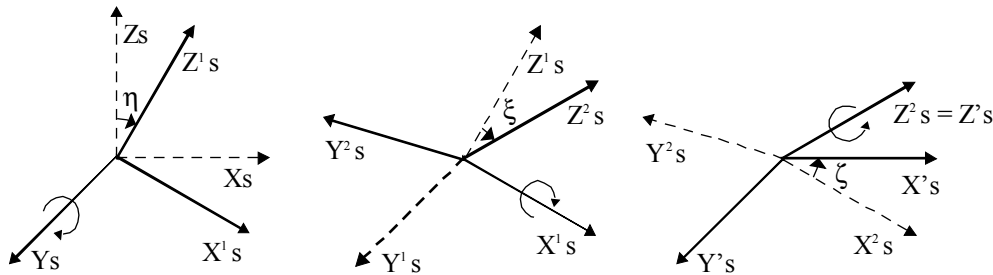
### 7.39.1 Overview

The `xl_euler_to_matrix` CFI function computes the rotation matrix equivalent to apply the three consecutive rotation through the given Euler angles. In other words, the result of multiplying the matrix to a vector is the same that applying the Euler rotations to the vector.

The rotation of a vector through the Euler angles is defined as three rotations of the reference frame:

1. Rotation around -Ys over a roll angle h
2. Rotation around -X1s (i.e the rotated Xs) over a pitch angle x
3. Rotation around +Z2s (i.e the rotated Z1s) over a yaw angle z.

Next drawing depicts the three rotations:



**Figure 4: Euler Angles**

#### Note on matrix notation:

If XYZ are the axes of the original reference frame, and X'Y'Z' are the axes of the rotated frame, the rows of the rotation matrix are respectively X, Y and Z axes expressed in X'Y'Z' system.

In the C representation, M[0][], M[1][], M[2][] are respectively 1st, 2nd and 3rd row of a rotation matrix M.

The rotation matrix M satisfies the following equivalence:

$$\mathbf{V} = \mathbf{M} \cdot \mathbf{V}'$$

where  $\mathbf{V}'$  is a vector expressed in the X'Y'Z' reference system and  $\mathbf{V}$  is expressed in the XYZ reference system.

### 7.39.2 Calling interface

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

```
#include <explorer_lib.h>
{
```

```

double angles[3];
double matrix[3][3];
long ierr[XL_NUM_ERR_EULER_TO_MATRIX], status;

status = xl_euler_to_matrix (angles, matrix, ierr);
}

```

The XL\_NUM\_ERR\_EULER\_TO\_MATRIX constant is defined in the file *explorer\_lib.h*.

### 7.39.3 Input parameters

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

**Table 139: Input parameters of `xl_euler_to_matrix` function**

| C name | C type    | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------|-----------|---------------|-------------------------|---------------|---------------|
| angles | double[3] | [0]           | Pitch angle             | degrees       | -             |
|        |           | [1]           | Roll angle              |               |               |
|        |           | [2]           | Yaw angle               |               |               |

### 7.39.4 Output parameters

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

**Table 140: Output parameters of `xl_euler_to_matrix` function**

| C name                          | C type        | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|---------------------------------|---------------|---------------|--|---------------|---------------|
| <code>xl_euler_to_matrix</code> | long          | -             | Status flag                                    | -             | -             |
| <code>matrix</code>             | double [3][3] | All           | Rotation matrix equivalent to the Euler angles | -             | -             |
| <code>ierr</code>               | long          | -             | Error vector                                   | -             | -             |

### 7.39.5 Warnings and errors

No errors have been envisaged for this function.

**Table 141: Runtime performances of `xl_euler_to_matrix` function**

| Solaris 32-bit.<br>[ms] | Solaris 64 bit.<br>[ms] | Linux 32-bit.<br>[ms] | Linux 64-bit.<br>[ms] |
|-------------------------|-------------------------|-----------------------|-----------------------|
| 0.00926                 | 0.00203                 | 0.00277               | 0.00124               |



Code: EO-MA-DMS-GS-0003  
Date: 06/02/12  
Issue: 4.3  
Page: 184

---



## 7.40xl\_matrix\_to\_euler

### 7.40.1Overview

The `xl_matrix_to_euler` CFI function computes the Euler angles (see section 7.39.1) equivalent to the input rotation matrix (the matrix is checked to be orthonormal; if not, an error is returned). This function is the inverse of `xl_euler_to_matrix`.

The transformation from a rotation matrix to Euler angles is not unique, there are two sets of angles that lead to the same rotation matrix. More precisely, the rotation given by (*pitch*, *roll*, *yaw*) is equivalent to (*180-pitch*, *180+roll*, *180+yaw*). Of the two possible solutions, this function chooses the one in which the *pitch* angle is between  $-90^\circ$  and  $+90^\circ$  (or  $\cos(\textit{pitch}) > 0$ )

Another indetermination happens when the pitch angle is  $\pm 90$ . In this case, the values for roll and yaw depends on each other. In this case the function returns a warning (section 7.40.5) and a solution is returned for which the yaw angle is set to 0.**Note on matrix notation:**

If XYZ are the axes of the original reference frame, and X'Y'Z' are the axes of the rotated frame, the rows of the rotation matrix are respectively X, Y and Z axes expressed in X'Y'Z' system.

In the C representation, `M[0][[]]`, `M[1][[]]`, `M[2][[]]` are respectively 1st, 2nd and 3rd row of a rotation matrix M.

The rotation matrix M satisfies the following equivalence:

$$\mathbf{V} = \mathbf{M} \cdot \mathbf{V}'$$

where  $\mathbf{V}'$  is a vector expressed in the X'Y'Z' reference system and  $\mathbf{V}$  is expressed in the XYZ reference system.

### 7.40.2Calling interface

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

```
#include <explorer_lib.h>
{
    double angles[3];
    double matrix[3][3];
    long ierr[XL_NUM_ERR_MATRIX_TO_EULER], status;

    status = xl_matrix_to_euler (matrix, angles, ierr);
}
```

The `XL_NUM_ERR_MATRIX_TO_EULER` constant is defined in the file `explorer_lib.h`.

### 7.40.3Input parameters

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

**Table 142: Input parameters of xl\_matrix\_to\_euler function**

| C name | C type        | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------|---------------|---------------|-------------------------|---------------|---------------|
| matrix | double [3][3] | All           | Rotation matrix         | -             | -             |

### 7.40.4 Output parameters

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

**Table 143: Output parameters of xl\_matrix\_to\_euler function**

| C name             | C type    | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|--------------------|-----------|---------------|-------------------------|---------------|---------------|
| xl_matrix_to_euler | long      | -             | Status flag             | -             | -             |
| angles             | double[3] | [0]           | Pitch angle             | degrees       | [-180, 180]   |
|                    |           | [1]           | Roll angle              |               |               |
|                    |           | [2]           | Yaw angle               |               |               |
| ierr               | long      | -             | Error vector            | -             | -             |

### 7.40.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_matrix_to_euler` CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 by translating the error vector returned by the `xl_matrix_to_euler` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 144: Error messages of xl\_matrix\_to\_euler function**

| Error type | Error message                         | Cause and impact   | Error code                                   | Error No |
|------------|---------------------------------------|--|--|----------|
| ERR        | The matrix does not define a rotation | No calculation performed   | XL_CFI_MATRIX_TO_EULER_WRONG_MATRIX_ERROR    | 0        |
| WARN       | Ambiguity in angles determination     | Calculation performed. The roll and yaw angles are indetermined, so the yaw is set to zero and the roll is computed as if the yaw were 0. In whatever case the three angles are equivalent to the rotation matrix. This situation happens when the pitch angle is 90° or -90°. | XL_CFI_MATRIX_TO_EULER_ANGLES_UNDEFINED_WARN | 1        |

---

|     |                               |   |  |   |
|-----|-------------------------------|---|--|---|
| ERR | The matrix is not orthonormal | No calculation performed. The CFI performs a check, with a tolerance of $10^{-6}$ , that the product of the input matrix and its transposed is the unitary matrix | XL_CFI_MATRIX_TO_EULER_ORTHONORMAL_ERR | 2 |
|-----|-------------------------------|---|--|---|

## 7.41 xl\_position\_on\_orbit

### 7.41.1 Overview

The `xl_position_on_orbit` CFI function calculates a value describing the position of the satellite within the orbit, using as input a Cartesian orbit state vector.

### 7.41.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long angle_type, time_ref, deriv;
    double time, pos[3], vel[3], acc[3],
    double angle, angle_rate, angle_rate_rate;
    xl_time_id time_id = {NULL};
    long status, ierr[XL_NUM_ERR_POSITION_ON_ORBIT];

    status = xl_position_on_orbit(&model_id,
                                &time_id,
                                &angle_type,
                                &time_ref, &time,
                                pos, vel, acc, &deriv,
                                &angle, &angle_rate,
                                &angle_rate_rate,
                                ierr);

    /* Or, using the run_id */
    long run_id;
    status = xl_position_on_orbit_run(&run_id,
                                     &angle_type,
                                     &time_ref, &time,
                                     pos, vel, acc, &deriv,
                                     &angle, &angle_rate,
                                     &angle_rate_rate,
                                     ierr);
}
```

### 7.41.3 Input parameters

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

*Table 145: Input parameters of `xl_position_on_orbit` function*

| C name                  | C type                    | Array Element | Description (Reference)                        | Unit (Format)                    | Allowed Range  |
|-------------------------|---------------------------|---------------|--|----------------------------------|--|
| <code>model_id</code>   | <code>xl_model_id*</code> | -             | Model ID                                       | -                                | -  |
| <code>time_id</code>    | <code>xl_time_id*</code>  | -             | Structure that contains the time correlations. | -                                | -  |
| <code>angle_type</code> | <code>long*</code>        | -             | Type of angle                                  | -                                | XL_ANGLE_TYPE_TRUE_LAT_TOD<br>XL_ANGLE_TYPE_MEAN_LAT_TOD             |
| <code>time_ref</code>   | <code>long*</code>        | -             | Time reference ID                              | -                                | Complete   |
| <code>time</code>       | <code>double*</code>      | -             | Reference time                                 | Decimal days (Processing format) | [-18262.0,36524.0]   |
| <code>pos</code>        | <code>double[3]</code>    | all           | Satellite position vector (Earth Fixed CS)     | m                                | -  |
| <code>vel</code>        | <code>double[3]</code>    | all           | Satellite velocity vector (Earth Fixed CS)     | m/s                              | -  |
| <code>acc</code>        | <code>double[3]</code>    | all           | Satellite acceleration vector (Earth Fixed CS) | m/s <sup>2</sup>                 | -  |
| <code>deriv</code>      | <code>long *</code>       | -             | Derivative ID                                  | -                                | Allowed values:<br>(0) XL_NO_DER<br>(1) XL_DER_1ST<br>(2) XL_DER_2ND |

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

- Time reference ID: `time_ref`.

### 7.41.4 Output parameters

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

*Table 146: Output parameters of `xl_position_on_orbit` function*

| C name  | C type               | Array Element | Description (Reference)                              | Unit (Format)      | Allowed Range |
|---|----------------------|---------------|--|--------------------|---------------|
| <code>angle</code>                              | <code>double*</code> | -             | Angle describing the position in the orbit           | deg                | -             |
| <code>angle_rate</code>                         | <code>double*</code> | -             | Angle describing the position in the orbit-rate      | deg/s              | -             |
| <code>angle_rate_rate</code>                    | <code>double*</code> | -             | Angle describing the position in the orbit-rate-rate | deg/s <sup>2</sup> | -             |
| <code>ierr[XL_NUM_ERR_POSITION_ON_ORBIT]</code> | <code>long</code>    | all           | Status vector  | -                  | -             |

### 7.41.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl\_position\_on\_orbit** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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, mainly on the results vector.

The table is completed by the error code and value. These error codes can be obtained translating the status vector returned by the **xl\_position\_on\_orbit** CFI function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 147: Error messages of xl\_position\_on\_orbit function**

| Error type | Error message                                   | Cause and impact         | Error code                                  | Error No |
|------------|---|--------------------------|---|----------|
| ERR        | Angle type is not valid                         | No calculation performed | XL_CFI_POSITION_ON_ORBIT_ANGLE_TYPE_ERR     | 0        |
| ERR        | Error occurred during call to xl_change_cart_cs | No calculation performed | XL_CFI_POSITION_ON_ORBIT_CHANGE_CART_CS_ERR | 1        |
| ERR        | Error occurred during call to XL_True_Lat       | No calculation performed | XL_CFI_POSITION_ON_ORBIT_TRUE_LAT_ERR       | 2        |

## 7.42 `xl_get_rotation_angles`

### 7.42.1 Overview

The `xl_get_rotation_angles` CFI function calculates the rotation angles between two sets of orthonormal right-handed unit vectors expressed wrt an identical coordinate frame.

### 7.42.2 Calling interface

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

```
#include <explorer_lib.h>
{
    double xs_initial[3], ys_initial[3], zs_initial[3];
    double xs_final[3], ys_final[3], zs_final[3];
    double ang[3];
    long ierr[XL_NUM_ERR_GET_ROTATION_ANGLES], status;
    status = xl_get_rotation_angles (xs_initial, ys_initial,
                                    zs_initial, xs_final, ys_final, zs_final,
                                    ang, ierr);
}
```

The `XL_NUM_ERR_GET_ROTATION_ANGLES` constant is defined in the file `explorer_lib.h`.

### 7.42.3 Input parameters

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

**Table 148: Input parameters of `xl_get_rotation_angles` function**

| C name                     | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|----------------------------|--------|---------------|---|---------------|---------------|
| <code>xs_initial[3]</code> | double | all           | Unitary direction vector along the X-axes of the initial attitude frame (Coordinate System) | -             | -             |
| <code>ys_initial[3]</code> | double | all           | Unitary direction vector along the Y-axes of the initial attitude frame (Coordinate System) | -             | -             |
| <code>zs_initial[3]</code> | double | all           | Unitary direction vector along the Z-axes of the initial attitude frame                     | -             | -             |

|             |        |     |   |   |   |
|-------------|--------|-----|---|---|---|
|             |        |     | (Coordinate System)   |   |   |
| xs_final[3] | double | all | Unitary direction vector along the X-axes of the final attitude frame (Coordinate System) | - | - |
| ys_final[3] | double | all | Unitary direction vector along the Y-axes of the final attitude frame (Coordinate System) | - | - |
| zs_final[3] | double | all | Unitary direction vector along the Z-axes of the final attitude frame (Coordinate System) | - | - |

### 7.42.4 Output parameters

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

**Table 149: Output parameters of `xl_get_rotation_angles` function**

| C name | C type | Array Element | Description (Reference)                               | Unit (Format) | Allowed Range |
|--------|--------|---------------|---|---------------|---------------|
| ang[3] | double | [0]           | Pitch angle between initial and final Attitude Frames | deg           | [-180,180)    |
|        |        | [1]           | Roll angle between initial and final Attitude Frames  | deg           | [-180,180)    |
|        |        | [2]           | Yaw angle between initial and final Attitude Frames   | deg           | [-180,180)    |
| ierr   | long   | -             | Error vector  | -             | -             |

### 7.42.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_get_rotation_angles` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_get_rotation_angles` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 150: Error messages of `xl_get_rotation_angles` function**

| Error type | Error message                    | Cause and impact  | Error code                                   | Error No |
|------------|----------------------------------|---|--|----------|
| ERR        | Input vectors are not orthogonal | No calculation performed. The CFI performs a check, with a tolerance of $10^{-6}$ , that the internal product of the input vectors is zero. | XL_CFI_GET_ROTATION_ANGLES_NO_ORTHOGONAL_ERR | 0        |



---

|     |   |                          |  |   |
|-----|---|--------------------------|--|---|
| ERR | Error occurred during call to function XL_CS_Rotation | No calculation performed | XL_CFI_GET_ROTATION_ANGLES_CS_ROTATION_ERR | 1 |
|-----|---|--------------------------|--|---|

## 7.43 `xl_get_rotated_vectors`

### 7.43.1 Overview

The `xl_get_rotated_vectors` CFI function calculates the rotated unit vectors given a set of unit vectors and the rotation angles expressed wrt an identical coordinate frame.

### 7.43.2 Calling interface

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

```
#include <explorer_lib.h>
{
    double xs_initial[3], ys_initial[3], zs_initial[3];
    double xs_final[3], ys_final[3], zs_final[3];
    double ang[3];
    long ierr[XL_NUM_ERR_GET_ROTATED_VECTORS], status;
    status = xl_get_rotated_vectors (xs_initial, ys_initial,
                                    zs_initial, ang, xs_final, ys_final,
                                    zs_final, ierr);
}
```

The `XL_NUM_ERR_GET_ROTATED_VECTORS` constant is defined in the file `explorer_lib.h`.

### 7.43.3 Input parameters

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

**Table 151: Input parameters of `xl_get_rotated_vectors` function**

| C name                     | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|----------------------------|--------|---------------|---|---------------|---------------|
| <code>xs_initial[3]</code> | double | all           | Unitary direction vector along the X-axes of the initial attitude frame (Coordinate System) |               |               |
| <code>ys_initial[3]</code> | double | all           | Unitary direction vector along the Y-axes of the initial attitude frame (Coordinate System) |               |               |
| <code>zs_initial[3]</code> | double | all           | Unitary direction vector along the Z-axes of the initial attitude frame (Coordinate System) |               |               |

|        |        |     |   |     |            |
|--------|--------|-----|---|-----|------------|
| ang[3] | double | [0] | Pitch angle between initial and final Attitude Frames | deg | [-180,180) |
|        |        | [1] | Roll angle between initial and final Attitude Frames  | deg | [-180,180) |
|        |        | [2] | Yaw angle between initial and final Attitude Frames   | deg | [-180,180) |

### 7.43.4 Output parameters

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

**Table 152: Output parameters of `xl_get_rotated_vectors` function**

| C name      | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|-------------|--------|---------------|---|---------------|---------------|
| xs_final[3] | double | all           | Unitary direction vector along the X-axes of the rotated attitude frame (Coordinate System) | -             | -             |
| ys_final[3] | double | all           | Unitary direction vector along the Y-axes of the rotated attitude frame (Coordinate System) | -             | -             |
| zs_final[3] | double | all           | Unitary direction vector along the Z-axes of the rotated attitude frame (Coordinate System) | -             | -             |
| ierr        | long   | -             | Error vector  | -             | -             |

### 7.43.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_get_rotated_vectors` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_get_rotated_vectors` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 153: Error messages of `xl_get_rotated_vectors` function**

| Error type | Error message                    | Cause and impact  | Error code                                   | Error No |
|------------|----------------------------------|---|--|----------|
| ERR        | Input vectors are not orthogonal | No calculation performed. The CFI performs a check, with a tolerance of $10^{-6}$ , that the internal product of the input vectors is zero. | XL_CFI_GET_ROTATED_VECTORS_NO_ORTHOGONAL_ERR | 1        |
| ERR        | Error occurred during call to    | No calculation performed  | XL_CFI_GET_ROTATED                           | 2        |



---

|  |                       |  |                           |  |
|--|-----------------------|--|---------------------------|--|
|  | function XL_Rotate_CS |  | VECTORS_ROTATE_C<br>S ERR |  |
|--|-----------------------|--|---------------------------|--|

## 7.44 **xl\_quaternions\_to\_vectors**

### 7.44.1 **Overview**

The `xl_quaternions_to_vectors` CFI function calculates the orthonormal unit vectors from a given set of quaternions.

### 7.44.2 **Calling interface**

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

```
#include <explorer_lib.h>
{
    double quaternions[4];
    double ux_vec[3], uy_vec[3], uz_vec[3];
    long ierr[XL_NUM_ERR_QUATERNIONS_TO_VEC], status;

    status = xl_quaternions_to_vectors (quaternions,
                                       ux_vec, uy_vec, uz_vec, ierr);
}
```

The `XL_NUM_ERR_QUATERNIONS_TO_VECTORS` constant is defined in the file *explorer\_lib.h*.

### 7.44.3 **Input parameters**

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

**Table 154: Input parameters of `xl_quaternions_to_vectors` function**

| C name      | C type    | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|-------------|-----------|---------------|-------------------------|---------------|---------------|
| quaternions | double[4] | -             | Quaternions             | -             | -             |

### 7.44.4 **Output parameters**

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

**Table 155: Output parameters of `xl_quaternions_to_vectors` function**

| C name                 | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|------------------------|--------|---------------|---|---------------|---------------|
| <code>ux_vec[3]</code> | double | all           | Unitary direction vector along the X-axes of the coordinate or attitude frame | -             | -             |
| <code>uy_vec[3]</code> | double | all           | Unitary direction vector along the Y-axes of the coordinate or attitude frame | -             | -             |
| <code>uz_vec[3]</code> | double | all           | Unitary direction vector along the Z-axes of the coordinate or attitude frame | -             | -             |
| <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 `xl_quaternions_to_vectors` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_quaternions_to_vectors` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 156: Error messages of `xl_quaternions_to_vectors` function**

| Error type | Error message  | Cause and impact         | Error code                                  | Error No |
|------------|--|--------------------------|---|----------|
| ERR        | Wrong input quaternion. The module is different from 1 | No calculation performed | XL_CFI_QUATERNIONS_TO_VEC_WRONG_INPUT_ERROR | 0        |

## 7.45 **xl\_vectors\_to\_quaternions**

### 7.45.1 **Overview**

The `xl_vectors_to_quaternions` CFI function calculates the set of quaternions that correspond to a set of orthonormal unit vectors.

### 7.45.2 **Calling interface**

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

```
#include <explorer_lib.h>
{
    double quaternions[4];
    double ux_vec[3], uy_vec[3], uz_vec[3];
    long ierr[XL_NUM_ERR_VEC_TO_QUATERNIONS], status;

    status = xl_vectors_to_quaternions (ux_vec, uy_vec, uz_vec,
                                       quaternions, ierr);
}
```

The `XL_NUM_ERR_VECTORS_TO_QUATERNIONS` constant is defined in the file `explorer_lib.h`.

### 7.45.3 **Input parameters**

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

**Table 157: Input parameters of `xl_vectors_to_quaternions` function**

| C name                 | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range |
|------------------------|--------|---------------|---|---------------|---------------|
| <code>ux_vec[3]</code> | double | all           | Unitary direction vector along the X-axes of the coordinate or attitude frame | -             | -             |
| <code>uy_vec[3]</code> | double | all           | Unitary direction vector along the Y-axes of the coordinate or attitude frame | -             | -             |
| <code>uz_vec[3]</code> | double | all           | Unitary direction vector along the Z-axes of the coordinate or attitude frame | -             | -             |

### 7.45.4 Output parameters

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

**Table 158: Output parameters of `xl_vectors_to_quaternions` function**

| C name      | C type    | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|-------------|-----------|---------------|-------------------------|---------------|---------------|
| quaternions | double[4] | -             | Quaternions             | -             | -             |
| ierr        | long      | -             | Error vector            | -             | -             |

### 7.45.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_vectors_to_quaternions` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_vectors_to_quaternions` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 159: Error messages of `xl_vectors_to_quaternions` function**

| Error type | Error message                     | Cause and impact  | Error code                                | Error No |
|------------|-----------------------------------|---|---|----------|
| ERR        | Wrong input vectors.              | No calculation performed  | XL_CFI_VEC_TO_QUATERNIONS_WRONG_INPUT_ERR | 0        |
| ERR        | Input vectors are not orthonormal | No calculation performed. The CFI performs a check, with a tolerance of $10^{-6}$ , that the internal product of the input vectors is zero. | XL_CFI_VEC_TO_QUATERNIONS_ORTHONORMAL_ERR | 1        |



## 7.46xl\_default\_sat\_init

### 7.46.1Overview

The `xl_default_sat_init` CFI function initializes a default satellite from a satellite configuration file (see [D\_H\_SUM]). This operation is needed whenever a default satellite is to be used for the first time, otherwise the satellite will not be recognized.

When the satellite is initialized, the function returns the satellite identifier (the `sat_id`). The `sat_id` cannot be chosen by the user, as the program will give the first available satellite if there is any. In order that a `sat_id` number can be used again for another initialization, it has to be freed by calling to the CFI function `xl_default_sat_close`.

**Important note:** Some parameters in the configuration file should be within the following ranges:

- nominal semimajor axis (a) >0
- nominal inclination (i): 0 deg < i < 180 deg
- nominal eccentricity (e): 10<sup>-6</sup> deg < e < 1

### 7.46.2Calling interface

The calling interface `xl_default_sat_init` function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long sat_id;
    char *conf_file;
    long ierr[XL_NUM_ERR_DEFAULT_SAT_INIT];
    long status;

    status = xl_default_sat_init(&sat_id, conf_file, ierr);
}
```

### 7.46.3Input parameters

The `xl_default_sat_init` function has the following input parameters:

**Table 160: Input parameters of `xl_default_sat_init` function**

| C name    | C type | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|-----------|--------|---------------|--|---------------|---------------|
| conf_file | char*  | -             | Path and name for the Satellite Configuration File (see [D_H_SUM] for further details about the configuration file). | -             | -             |

### 7.46.4 Output parameters

The output parameters of the `xl_default_sat_init` function are:

**Table 161: Output parameters of `xl_default_sat_init` function**

| C name | C type | Array Element | Description (Reference)   | Unit (Format) | Allowed Range                          |
|--------|--------|---------------|---|---------------|--|
| sat_id | long*  | -             | Satellite ID<br>The value is assigned automatically if there is an available satellite. | -             | From XL_SAT_DEFAULT to XL_SAT_DEFAULT9 |
| ierr   | long*  | all           | Error status flags  | -             |  |

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

- Satellite ID: `sat_id`. See [GEN\_SUM].

### 7.46.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_default_sat_init` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_default_sat_init` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

**Table 162: Error messages of `xl_default_sat_init` function**

| Error type | Error message                                    | Cause and impact  | Error code                            | Error No |
|------------|--|---|---------------------------------------|----------|
| ERR        | Default satellite ID is not correct              | The satellite identification number does not belong to a default satellite.<br>No computation performed | XL_CFI_DEFAULT_SAT_INIT_SAT_ERR       | 0        |
| ERR        | Error while reading satellite configuration file | Wrong configuration file.<br>No computation performed   | XL_CFI_DEFAULT_SAT_INIT_READ_FILE_ERR | 1        |

## 7.47 xl\_default\_sat\_close

### 7.47.1 Overview

The `xl_default_sat_close` CFI function frees a default satellite id. that was initialized with `xl_default_sat_init`, so that it can be used again.

### 7.47.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long sat_id;
    xl_default_sat_close(&sat_id);
}
```

### 7.47.3 Input parameters

The `xl_default_sat_close` function has the following input parameters:

*Table 163: Input parameters of xl\_default\_sat\_close function*

| C name | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range                          |
|--------|--------|---------------|-------------------------|---------------|--|
| sat_id | long*  | -             | Satellite ID to free.   | -             | From XL_SAT_DEFAULT to XL_SAT_DEFAULT9 |

### 7.47.4 Output parameters

This function does not return any value nor parameters.

### 7.47.5 Warnings and errors

No warning nor errors are returned

## 7.48xl\_set\_tle\_sat\_data

### 7.48.10overview

The `xl_set_tle_sat_data` CFI function changes the NORAD data that has been pre-established for every pre-defined satellite. These data are used for the reading/writing of the TLE files. The NORAD data that are defined for every satellite are:

- NORAD Satellite number
- NORAD Satellite name
- NORAD International designator

This data can be found in: <http://celestrak.com/satcat/search.asp>

This function should not be needed if the NORAD data were already stabilised for every Earth Observation satellite, however the data are not set until the satellite launch. So, in order to be able to handle TLE files it could be necessary to call this function so set the correct data for the satellite.

The Table 165 shows the list of predefined satellites in the Earth Observation CFI. Those satellites for which the NORAD data are not known has a NORAD Satellite number equal to "00000".

For the default satellites, that NORAD data cannot be set with this function as the data are set in the satellite configuration file.

WARNING: This function is not thread safe. However, this function could be used safely in the following way:

- The function should be called only if the real NORAD data are not correctly defined in the Table 165.
- The function should be called only if TLE files are going to be handled to generate TLE files or for orbit initialization
- The function should be called only once and at the beginning of a program before launching any thread.

**Table 164: NORAD Identifiers for satellites**

| Satellite ID     | NORAD Satellite Number | NORAD Satellite Name | NORAD International Designator |
|------------------|------------------------|----------------------|--------------------------------|
| XD SAT ERS1      | 21574                  | ERS1                 | "91050A "                      |
| XD SAT ERS2      | 23560                  | ERS2                 | "95021A "                      |
| XD SAT ENVISAT   | 27386                  | ENVISAT              | "02009A "                      |
| XD SAT METOP1    | 29499                  | METOP-A              | "06044A "                      |
| XD SAT METOP2    | 00000                  | METOP-B              | "00000 "                       |
| XD SAT METOP3    | 00000                  | METOP-C              | "00000 "                       |
| XD SAT CRYOSAT   | 36508                  | CRYOSAT 2            | "10013A "                      |
| XD SAT ADM       | 00000                  | AEOLUS               | "00000 "                       |
| XD SAT GOCE      | 34602                  | GOCE                 | "09013A "                      |
| XD SAT SMOS      | 36036                  | SMOS                 | "09059A "                      |
| XD SAT TERRASAR  | 00000                  | TERRASAR             | "00000 "                       |
| XD SAT EARTHCARE | 00000                  | EARTHCARE            | "00000 "                       |
| XD SAT SWARM A   | 00000                  | SWARM-A              | "00000 "                       |
| XD SAT SWARM B   | 00000                  | SWARM-B              | "00000 "                       |
| XD_SAT_SWARM_C   | 00000                  | SWARM-C              | "00000 "                       |

|                    |       |            |          |
|--------------------|-------|------------|----------|
| XD SAT SENTINEL 1A | 00000 | SENTINEL1A | "00000 " |
| XD SAT SENTINEL 1B | 00000 | SENTINELB  | "00000 " |
| XD SAT SENTINEL 2  | 00000 | SENTINEL2  | "00000 " |
| XD SAT SENTINEL 3  | 00000 | SENTINEL3  | "00000 " |
| XD SAT SENTINEL 1C | 00000 |            |          |
| XD SAT SENTINEL 2A | 00000 |            |          |
| XD SAT SENTINEL 2B | 00000 |            |          |
| XD SAT SENTINEL 2C | 00000 |            |          |
| XD SAT SENTINEL 3A | 00000 |            |          |
| XD SAT SENTINEL 3B | 00000 |            |          |
| XD SAT SENTINEL 3C | 00000 |            |          |
| XD SAT SEOSAT      | 00000 | SEOSAT     | "00000 " |
| XD SAT GENERIC     | 00000 | GENERIC    | "00000 " |

### 7.48.2 Calling interface

The calling interface `xl_set_tle_sat_data` function is the following (input parameters are underlined):

```
#include <explorer_lib.h>
{
    long sat_id, status;
    long norad_sat_number;
    char norad_satcat[25];
    char int_des[9];
    status = xl_set_tle_sat_data (&sat_id,
                                &norad_sat_number,
                                norad_satcat,
                                int_des);
}
```

### 7.48.3 Input parameters

The `xl_set_tle_sat_data` function has the following input parameters:

*Table 165: Input parameters of `xl_set_tle_sat_data` function*

| C name | C type | Array Element | Description (Reference) | Unit (Format) | Allowed Range                               |
|--------|--------|---------------|-------------------------|---------------|---|
| sat_id | long*  | -             | Satellite ID            | -             | Any predefined satellite shown in Table 164 |

### 7.48.4 Output parameters

This function returns the status of the execution:

- 0 if the execution was correct
- -1 if an error occurred. This only could happen if the input sat\_id was incorrect.

### ***7.48.5 Warnings and errors***

No warning nor errors are returned

## 7.49xl\_model\_init

### 7.49.10 Overview

The `xl_model_init` CFI function initialises the model id with the requested models. There are two ways to initialise the `model_id`:

- Selecting a set of models via an enumeration value.
- Selecting a specific model for every model types.

Note that the `model_id` can be used if it has not been initialised. In that case, the default CFI models are used.

The following table shows the possible models for every model type:

**Table 166: Possible models for every model type**

| Model type  | Models                                   |
|---|--|
| Earth model<br>( <code>XL_MODEL_TYPE_EARTH</code> )           | <code>XL_MODEL_EARTH_DEFAULT</code>      |
| Sun model<br>( <code>XL_MODEL_TYPE_SUN</code> )               | <code>XL_MODEL_SUN_DEFAULT</code>        |
| Moon model<br>( <code>XL_MODEL_TYPE_MOON</code> )             | <code>XL_MODEL_MOON_DEFAULT</code>       |
| Planet model<br>( <code>XL_MODEL_TYPE_PLANET</code> )         | <code>XL_MODEL_PLANET_DEFAULT</code>     |
| Star model<br>( <code>XL_MODEL_TYPE_STAR</code> )             | <code>XL_MODEL_STAR_DEFAULT</code>       |
| Nutation model<br>( <code>XL_MODEL_TYPE_NUTATION</code> )     | <code>XL_MODEL_NUTATION_DEFAULT</code>   |
| Precession model<br>( <code>XL_MODEL_TYPE_PRECESSION</code> ) | <code>XL_MODEL_PRECESSION_DEFAULT</code> |
| Constants model<br>( <code>XL_MODEL_TYPE_CONSTANTS</code> )   | <code>XL_MODEL_CONSTANTS_DEFAULT</code>  |

**Table 167: Model sets**

| Model set                     | Selected Models   |
|-------------------------------|---|
| <code>XL_MODEL_DEFAULT</code> | <code>XL_MODEL_EARTH_DEFAULT</code> ,<br><code>XL_MODEL_SUN_DEFAULT</code> ,<br><code>XL_MODEL_MOON_DEFAULT</code> ,<br><code>XL_MODEL_PLANET_DEFAULT</code> ,<br><code>XL_MODEL_STAR_DEFAULT</code> ,<br><code>XL_MODEL_NUTATION_DEFAULT</code> ,<br><code>XL_MODEL_PRECESSION_DEFAULT</code> ,<br><code>XL_MODEL_CONSTANTS_DEFAULT</code> |
| <code>XL_MODEL_CONFIG</code>  | The models are chosen by the user with the models from Table 166  |

In order to simplify the the initialisation, it is possible to select a set of models to be used.

### 7.49.2 Calling interface

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

```
#include <explorer_lib.h>
{
    long mode, models[XL_NUM_MODEL_TYPES_ENUM];
    xl_model_id model_id = {NULL};
    long ierr[XL_NUM_ERR_MODEL_INIT], status;

    status = xl_model_init (&mode, models,
                           &model_id,
                           ierr)
}
```

The `XL_NUM_MODEL_TYPES_ENUM` and `XL_NUM_ERR_MODEL_INIT` constant is defined in the file `explorer_lib.h`.

### 7.49.3 Input parameters

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

**Table 168: Input parameters of `xl_model_init` function**

| C name | C type | Array Element | Description (Reference)  | Unit (Format) | Allowed Range |
|--------|--------|---------------|--|---------------|---------------|
| mode   | long   | -             | model set (according to Table 167)   | -             | -             |
| models | long[] | all           | These models are used in case of setting the <i>mode</i> parameter to <code>XL_MODEL_CONFIG</code> . The models are defined in Table 166 | -             | -             |
|        |        | 0             | Earth model  | -             | -             |
|        |        | 1             | Sun model  |               |               |
|        |        | 2             | Moon model   |               |               |
|        |        | 3             | Planet model   |               |               |
|        |        | 4             | Star model   |               |               |
|        |        | 5             | Nutation model   |               |               |
|        |        | 6             | Precession model   |               |               |
|        |        | 7             | Constants model  |               |               |



### 7.49.4 Output parameters

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

*Table 169: Output parameters of `xl_model_init` function*

| C name                     | C type                   | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|----------------------------|--------------------------|---------------|-------------------------|---------------|---------------|
| <code>xl_model_init</code> | long                     | -             | Status flag             | -             | -             |
| <code>model_id</code>      | <code>xl_model_id</code> | -             | Model ID                | -             | -             |
| <code>ierr</code>          | long*                    | all           | Error array             | -             | -             |

### 7.49.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_model_init` CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library `xl_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 extended status flag returned by the `xl_model_init` function by calling the function of the EO\_LIB software library `xl_get_code` (see [GEN\_SUM]).

*Table 170: Error messages of `xl_model_init` function*

| Error type | Error message                        | Cause and impact         | Error code                        | Error No |
|------------|--------------------------------------|--------------------------|-----------------------------------|----------|
| ERR        | Model ID is already initialised      | No calculation performed | XL_CFI_MODEL_INIT_STATUS_ERR      | 0        |
| ERR        | Memory allocation error              | No calculation performed | XL_CFI_MODEL_INIT_MEMORY_ERR      | 1        |
| ERR        | Wrong enumeration value for %s model | No calculation performed | XL_CFI_MODEL_INIT_WRONG_MODEL_ERR | 2        |

## 7.50 `xl_model_close`

### 7.50.1 Overview

The `xl_model_close` CFI function cleans up any memory allocation performed by the initialization functions.

A complete calling sequence of the time reference computations is presented in section 4.2.

### 7.50.2 Calling interface

The calling interface of the `xl_model_close` CFI function is the following:

```
#include <explorer_lib.h>
{
    xl_model_id model_id = {NULL};
    long ierr[XL_NUM_ERR_MODEL_CLOSE], status;
    status = xl_model_close (&model_id, ierr);
}
```

### 7.50.3 Input parameters

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

*Table 171: Input parameters of `xl_model_close` function*

| C name                | C type                    | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|-----------------------|---------------------------|---------------|--|---------------|---------------|
| <code>model_id</code> | <code>xl_model_id*</code> | -             | Structure that contains the time correlations. | -             | -             |

### 7.50.4 Output parameters

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

*Table 172: Output parameters of `xl_model_close` function*

| C name                      | C type            | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|-----------------------------|-------------------|---------------|-------------------------|---------------|---------------|
| <code>xl_model_close</code> | <code>long</code> | -             | Status flag             | -             | -             |
| <code>ierr</code>           | <code>long</code> | -             | Error vector            | -             | -             |

### 7.50.5 Warnings and errors

Next table lists the possible error messages that can be returned by the `xl_model_close` CFI function after

translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO\_LIB software library **xl\_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 extended status flag returned by the **xl\_model\_close** function by calling the function of the EO\_LIB software library **xl\_get\_code** (see [GEN\_SUM]).

**Table 173: Error messages of xl\_model\_close function**

| <b>Error type</b> | <b>Error message</b>   | <b>Cause and impact</b>  | <b>Error code</b>               | <b>Error No</b> |
|-------------------|--|--------------------------|---------------------------------|-----------------|
| ERR               | The Model Id is not initialized or it could be in use by another Id. | No calculation performed | XL_CFI_MODEL_CLOSE_WRONG_ID_ERR | 0               |

## 7.51 xl\_model\_get\_data

### 7.51.1 Overview

The `xl_model_get_data` CFI function returns a data structure containing the data used for the time initialisation.

### 7.51.2 Calling interface

The calling interface of the `xl_model_get_data` CFI function is the following:

```
#include <explorer_lib.h>
{
    xl_model_id model_id;
    xl_model_id_data data;
    long status;
    status = xl_model_get_data (&model_id, &data);
}
```

### 7.51.3 Input parameters

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

*Table 174: Input parameters of xl\_model\_get\_data function*

| C name   | C type       | Array Element | Description (Reference)                        | Unit (Format) | Allowed Range |
|----------|--------------|---------------|--|---------------|---------------|
| model_id | xl_model_id* | -             | Structure that contains the model information. | -             | -             |

### 7.51.4 Output parameters

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

*Table 175: Output parameters of xl\_model\_get\_data function*

| C name            | C type        | Array Element | Description (Reference) | Unit (Format) | Allowed Range |
|-------------------|---------------|---------------|-------------------------|---------------|---------------|
| xl_model_get_data | long          | -             | Status flag             | -             | -             |
| data              | xl_model_data | -             | Model ID data           | -             | -             |

The data structure `xl_model_data` can be seen in Table 8.

### **7.51.5 Warnings and errors**

This function does not return any error/warning code. Only the status of the function indicates if the execution was correct or not.

The possible causes of error are:

- The model\_id was not initialised.

## 8 CFI EXECUTABLE PROGRAMS

The following sections describe executables programs based on the CFI functions.

### 8.1 time\_conv

This program makes time conversions between different formats and time references. It is call in the following way:

```
time_conv    [-ref_in] input_time_ref
             [-ref_out] output_time_ref
             -fmt_in input_format
             -fmt_out output_format
             {-day j200_date (days)|
             (-t1 tranport_1 [-t2 transport_2] [-t3 transport_3] [-t4 transport_4]) |
             -date string_date (date)}
             [-v ]
             [-xd_v ]
             [-xl_v ]
             [-help ]
             [-show]
             [{ (-tai TAI_time -gps GPS_time -utc UTC_time -ut1 UT1_time) |
             (-tmod time_model -tfile time_file -trid time_reference
             {(-tm0 time0 -tm1 time1) | (-orb0 orbit0 -orb1 orbit1) } ) } ]
```

Note that:

- Order of parameters does not matter.
- Bracketed parameters are not mandatory.
- Options between curly brackets and separated by a vertical bar are mutually exclusive.
- [ -xd\_v ] option for EO\_DATA\_HANDLING Verbose mode.
- [ -xl\_v ] option for EO\_LIB Verbose mode.
- [ -v ] option for Verbose mode for all libraries (default is Silent).
- [ -show ] displays the inputs of the function and the results.
- Possible values for time\_model: USER, NONE , IERS\_B\_PREDICTED, IERS\_B\_RESTITUTED, FOS\_PREDICTED, FOS\_RESTITUTED, DORIS\_PRELIMINARY, DORIS\_PRECISE, DORIS\_NAVIGATOR.
- Possible values for time\_ref and time\_reference: TAI, UTC, UT1, GPS.

- Possible values for input\_format and output\_format:
  - Julian days: PROC
  - Transport format: TRANS\_STD, TRANS\_ENVI\_GS, TRANS\_CRYO\_GS, TRANS\_CRYO\_TM, TRANS\_CRYO\_TM\_SIRAL, SMOS\_TM
  - date string: ASCII\_STD, ASCII\_STD\_REF, ASCII\_STD\_MICROSEC, ASCII\_STD\_REF\_MICROSEC, ASCII\_COMPACT, ASCII\_COMPACT\_REF, ASCII\_COMPACT\_MICROSEC, ASCII\_COMPACT\_REF\_MICROSEC, ASCII\_ENVI, ASCII\_ENVI\_REF, ASCII\_ENVI\_MICROSEC, ASCII\_ENVI\_REF\_MICROSEC, ASCII\_CCSDSA, ASCII\_CCSDSA\_REF, ASCII\_CCSDSA\_MICROSEC, ASCII\_CCSDSA\_REF\_MICROSEC, ASCII\_CCSDSA\_COMPACT, ASCII\_CCSDSA\_COMPACT\_REF, ASCII\_CCSDSA\_COMPACT\_MICROSEC, ASCII\_CCSDSA\_COMPACT\_REF\_MICROSEC
- The last three lines of parameters are used for initialising the time correlations . Note that only one set of parameters should be introduced
  - TAI, GPS, UTC and UT1 input times (as in xl\_time\_ref\_init)
  - A file with time reference data, the time mode, the time reference name and a time range (as in xl\_time\_ref\_init\_file)
- In a time conversion, if the time reference is not to be changed, the values for "-ref\_in", "-ref\_out" and the parameters for the time initialization are not needed. Note that the time reference will be always requested if the input/output format contains the reference in the date.

Examples:

```
time_conv -t1 1550 -t2 44266 -t3 176000 -t4 0 -fmt_in TRANS_STD
          -fmt_out PROC -ref_in TAI -ref_out TAI -v
          -tai 245.100000000000 -gps 245.099780092
          -utc 245.099594907407 -ut1 245.099587962
```

```
time_conv -date 2004-03-30T12:17:46.176000
          -fmt_in ASCII_CCSDSA_MICROSEC
          -fmt_out ASCII_STD -ref_in TAI -ref_out GPS -v
          -tai 245.100000000000 -gps 245.099780092
          -utc 245.099594907407 -ut1 245.099587962
```

## 9 RUNTIME PERFORMANCES

The library performance has been measured by dedicated test procedures run in 5 different platforms under the below specified machines:

| <i>OS ID</i>   | <i>Processor</i>                            | <i>OS</i>   | <i>RAM</i> |
|----------------|---|---|------------|
| LINUX64        | Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz | GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)                     | 4 GB       |
| LINUX32_LEGACY | Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz | GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)                     | 4 GB       |
| LINUX64_LEGACY | Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz | GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)                     | 4 GB       |
| MACIN64        | Intel(R) Core(TM)2 Duo @ 2.00GHz            | MAC OS X V10.5.8  | 1 GB       |
| WINDOWS        | Intel(R) Core(TM)2 Duo CPU T8100 @ 2.10 GHz | Microsoft Windows XP Professional Version 2002 Service Pack 3 | 2.99 GB    |

The table below shows the time (in milliseconds - ms) each function takes to be run under each platform:

| <b>Function_id</b>   | <b>LINUX64</b> | <b>LINUX32_LEGACY</b> | <b>LINUX64_LEGACY</b> | <b>MACIN64</b> | <b>WINDOWS</b> |
|--|----------------|-----------------------|-----------------------|----------------|----------------|
| xl_time_ref_init_file *<br>File read: BULLETIN B 169. 874 records read. Depends on file reading implemented in explorer_data_handling. | 2.000000       | 2.200000              | 2.400000              | 5.400000       | 42.799999      |
| xl_time_ref_init   | 0.000850       | 0.001250              | 0.000860              | 0.002450       | 0.004437       |
| xl_change_cart_cs  | 0.004010       | 0.008150              | 0.004720              | 0.010810       | 0.012012       |
| xl_cart_to_geod  | 0.003220       | 0.006040              | 0.003270              | 0.007310       | 0.007420       |
| xl_geod_to_cart  | 0.000550       | 0.000950              | 0.000540              | 0.001120       | 0.001047       |
| xl_cart_to_kepl  | 0.000710       | 0.001250              | 0.000750              | 0.002620       | 0.001453       |
| xl_kepl_to_cart  | 0.000840       | 0.001520              | 0.001000              | 0.002090       | 0.001937       |
| xl_geod_distance   | 0.008000       | 0.012000              | 0.009000              | 0.022000       | 0.023400       |
| xl_sun   | 0.014000       | 0.024000              | 0.015000              | 0.033000       | 0.032800       |
| xl_moon  | 0.016000       | 0.029000              | 0.024000              | 0.049000       | 0.035900       |
| xl_planet  | 0.016000       | 0.028000              | 0.018000              | 0.039000       | 0.042200       |
| xl_star_radec  | 0.014000       | 0.023000              | 0.014000              | 0.031000       | 0.034300       |
| xl_time_transport_to_ascii   | 0.002970       | 0.004650              | 0.003390              | 0.008100       | 0.005358       |
| xl_time_transport_to_transport   | 0.001030       | 0.001830              | 0.001130              | 0.001940       | 0.001921       |
| xl_time_transport_to_processing  | 0.001010       | 0.001840              | 0.001140              | 0.001930       | 0.001812       |



|   |          |          |          |          |          |
|---|----------|----------|----------|----------|----------|
| xl_time_processing_to_ascii                               | 0.003150 | 0.005010 | 0.003410 | 0.008330 | 0.005702 |
| xl_time_processing_to_transport                           | 0.000890 | 0.001740 | 0.001040 | 0.001630 | 0.001656 |
| xl_time_processing_to_processing                          | 0.000890 | 0.001740 | 0.000900 | 0.001650 | 0.001671 |
| xl_time_ascii_to_ascii                                    | 0.004250 | 0.006540 | 0.004800 | 0.012770 | 0.007764 |
| xl_time_ascii_to_transport                                | 0.002100 | 0.003620 | 0.002340 | 0.005960 | 0.003764 |
| xl_time_ascii_to_processing                               | 0.004180 | 0.007520 | 0.004670 | 0.011910 | 0.007560 |
| xl_time_get_leap_second_info                              | 0.004490 | 0.007920 | 0.004990 | 0.012470 | 0.008169 |
| xl_time_init_status                                       | 0.000010 | 0.000010 | 0.000000 | 0.000020 | 0.000016 |
| xl_time_get_sat_id  | 0.000004 | 0.000007 | 0.000004 | 0.000011 | 0.000009 |
| xl_time_get_mode  | 0.000003 | 0.000009 | 0.000003 | 0.000011 | 0.000009 |
| xl_time_get_id  | 0.000110 | 0.000150 | 0.000110 | 0.000350 | 0.000453 |
| xl_time_set_id  | 0.000090 | 0.000130 | 0.000100 | 0.000370 | 0.000375 |
| xl_time_add   | 0.000310 | 0.000410 | 0.000310 | 0.000550 | 0.000329 |
| xl_time_diff  | 0.000460 | 0.000630 | 0.000460 | 0.000810 | 0.000484 |
| xl_time_obt_to_time                                       | 0.000490 | 0.000790 | 0.000490 | 0.000870 | 0.000578 |
| xl_time_time_to_obt                                       | 0.000630 | 0.000860 | 0.000640 | 0.001150 | 0.000718 |
| xl_get_rotation_angles                                    | 0.000740 | 0.001830 | 0.000710 | 0.002030 | 0.002906 |
| xl_get_rotated_vectors                                    | 0.000670 | 0.001560 | 0.001110 | 0.001520 | 0.001953 |
| xl_position_on_orbit                                      | 0.006050 | 0.011480 | 0.006730 | 0.014530 | 0.018792 |
| xl_cart_to_radec  | 0.000630 | 0.000880 | 0.000740 | 0.001490 | 0.001218 |
| xl_radec_to_cart  | 0.000600 | 0.000950 | 0.000620 | 0.001140 | 0.001094 |
| xl_euler_to_matrix  | 0.000360 | 0.000610 | 0.000930 | 0.000720 | 0.000843 |
| xl_matrix_to_euler  | 0.000380 | 0.000660 | 0.000370 | 0.001050 | 0.001500 |
| xl_star_catalog   | 0.032380 | 0.054480 | 0.035820 | 0.073240 | 0.087946 |
| xl_topocentric_to_ef                                      | 0.004800 | 0.007700 | 0.004800 | 0.010400 | 0.012650 |
| xl_ef_to_topocentric                                      | 0.004900 | 0.007900 | 0.005200 | 0.010700 | 0.013280 |
| xl_vectors_to_quaternions                                 | 0.000300 | 0.000400 | 0.000300 | 0.000700 | 0.001090 |
| xl_quaternions_to_vectors                                 | 0.000200 | 0.000300 | 0.000300 | 0.000400 | 0.000470 |
| xl_default_sat_init *<br>called with xl_default_sat_close | 0.000300 | 0.000500 | 0.000400 | 0.000600 | 0.000780 |

Note that when the value “0.000000” is defined for a function in a certain platform, it means that its running time is lower than 1 nano-second and so it can be considered as “0”.

## 10 LIBRARY PRECAUTIONS

The following precaution shall be taking into account when using EO\_LIB library:

- When a message like:

<LIBRARY NAME> >>> ERROR in *xl\_function*: Internal computation error # n

or

<LIBRARY NAME> >>> WARNING in *xl\_function*: Internal computation warning # n

appears, run the program in **verbose** mode for a complete description of warnings and errors and call for maintenance if necessary.

## 11 KNOWN PROBLEMS

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

*Table 176: Known problems*

| CFI library           | Problem   | Work around solution |
|-----------------------|---|----------------------|
| xl_time_obt_to_time   | Cannot be called from ForTran<br>(no ForTran equivalence to unsigned long)  | -                    |
| xl_time_obt_to_time   | Since OBT is presently defined only for Envisat and GOCE, this function only works if:<br>sat_id=XL_SAT_ENVISAT, sat_id=XL_SAT_GOCE,<br>sat_id=XL_SAT_ADM or<br>sat_id=XL_SAT_SMOS  | -                    |
| xl_time_time_to_obt   | Cannot be called from ForTran<br>(no ForTran equivalence to unsigned long)  | -                    |
| xl_time_time_to_obt   | Since OBT is presently defined only for Envisat and GOCE, this function only works if:<br>sat_id=XL_SAT_ENVISAT,<br>sat_id=XL_SAT_GOCE,<br>sat_id=XL_SAT_ADM or<br>sat_id=XL_SAT_SMOS   | -                    |
| xl_euler_to_matrix    | Functionality is not currently available  | -                    |
| xl_matrix_to_euler    | Functionality is not currently available  | -                    |
| xl_time_ref_init_file | XL_SEL_FILE is not supported for time inputs selection (i.e. it is required to specify a time or an orbit range).   | -                    |
| xl_change_cart_cs     | Changes from/to B1950 or GALACTIC coordinates are not currently available.  | -                    |
| xl_cart_radec         | Functionality is not currently available  | -                    |
| xl_radec_to_cart      | Functionality is not currently available  | -                    |
| xl_star_catalog       | Functionality is not currently available.   | -                    |
| xl_topocentric_to_ef  | Functionality is not currently available  | -                    |
| xl_ef_to_topocentric  | Functionality is not currently available  | -                    |
| All                   | The error message "Satellite ID is not correct" that appears in some error tables in section 7 is obsolete. It is highly recommended using the enumeration values for the error (error codes) instead of the integer value (error number) to identify the error messages as these error codes will be removed in the next delivery. | -                    |