

indra

Indra Sistemas S.A.
Mar Egeo 4 • Polígono Industrial nº 1
SAN FERNANDO DE HENARES
28830 MADRID
Tel. +34 916 273 001
Fax +34 916 273 051

Ref.: SO-RN-IDR-GS-0256
Iss./Rev.: 1/0
Date: 22-Nov-2017

SMOS DPGS

XML R/W API Release Note v05-00-00

	Name	Signature	Date
Prepared	M. Moreno		22-Nov-2017
Checked	M. Zapata		
Accepted	J. Ortega		
Authorized	M. Rodríguez		

File: SO-RN-IDR-GS-0256 XML RW API Release Note v05-00-00 2017-11-22

Document classification



ESA Signature:

Not needed

Not needed



indra



Ref. : SO-RN-IDR-GS-0256
 Iss./Rev. : 1/0
 Date: 22-Nov-2017

Distribution List

Internal	Copies	External	Copies
Archive	1	ESA	1

Document Change Log

Iss./Rev.	Date	Section / Page	Change Description
1/0	22-Nov-2017	All	Issue for XML R/W API version 05-00-00 delivery. This delivery includes the new version of XML R/W API for Centos 7.

Contents

1. INTRODUCTION.....	4
1.1 OBJECTIVE	4
1.2 ACRONYMS AND TERMS	4
1.3 DOCUMENT OVERVIEW	4
2. DOCUMENTS.....	5
2.1 APPLICABLE DOCUMENTS.....	5
3. XML READ/WRITE API SW OVERVIEW	6
3.1 OVERVIEW	6
3.2 SYSTEM REQUIREMENTS	9
3.2.1 Linux	9
3.2.1.1 Hardware Requirements	9
3.2.1.2 Software Requirements.....	9
3.2.2 Windows XP.....	10
3.2.2.1 Hardware Requirements	10
3.2.2.2 Software Requirements.....	10
3.2.3 MacOS X.....	10
3.2.3.1 Hardware Requirements	10
3.2.3.2 Software Requirements.....	11
3.2.4 Cygwin	11
3.2.4.1 Hardware Requirements	11
3.2.4.2 Software Requirements.....	11
3.2.5 Solaris	12
3.2.5.1 Hardware Requirements	12
3.2.5.2 Software Requirements.....	12
4. CONTENTS OF THE DELIVERY.....	13
5. APPLICABLE XML SCHEMAS LIST.....	15
6. CHANGE LIST.....	16

List of Figures

Figure 3-1: XML Read/Write API Library Interfaces	6
Figure 3-2: API interface collaboration diagram	7
Figure 3-3: API interface context diagram.....	8
Figure 3-4: Java interface	8

List of Tables

Table 2-1: Applicable Documents	5
---------------------------------------	---



1. INTRODUCTION

1.1 OBJECTIVE

The aim of this document is to provide the configuration status of the XML Read/Write API SW. The XML Read/Write API is a library to access Science SMOS products in Earth Explorer format. It will be mainly used by the developers of the SMOS components but it is designed and presented as a general public library. Moreover the XML Read/Write API is designed to fulfil the multi platform requirements.

This document contains the following information related to this release of the XML R/W API:

- Baseline documentation applicable to the SW release
- Description Contents of the XML R/W API SW 05-00-00
- Description of the means and tools necessary to develop, modify, generate and run the SW
- Installation instructions
- List of changes affecting the SW

IMPORTANT NOTE:

THIS DELIVERY ONLY WORKS WITH SCHEMAS VERSION 2007-11-19_v03-02-00 and later.

1.2 ACRONYMS AND TERMS

Refer to document SO-TN-IDR-GS-0010 [AD.1.].

1.3 DOCUMENT OVERVIEW

The XML R/W API Release Note has been organised as follows:

- Section 1, the section you are currently reading
- Section 2, applicable and reference documents
- Section 3, XML R/W API SW overview
- Section 4, Content of the delivery
- Section 5, Applicable XML Schemas list
- Section 6, Changes list



2. DOCUMENTS

2.1 APPLICABLE DOCUMENTS

The following documents are fully applicable to the SMOS DPGS implementation:

Ref.	Title	Code	Version	Date
[AD.1.]	SMOS DPGS Acronyms	SO-TN-IDR-GS-0010	2.1	29-Feb-2016
[AD.2.]	XML R/W API Software User Manual	SO-ID-IDR-GS-0009	2.2	09-Jul-2010
[AD.3.]	Explorer_Data_Handling SUM	CS-MA-DMS-GS-0009	3.5	07-Jul-2006
[AD.4.]	XML Schema Guidelines	SO-MA-IDR-GS-0004	2.1	09-Jul-2010
[AD.5.]	DPGS Schemas Versioning	SO-TN-IDR-GS-0024	1.7	31-Mar-2014

Table 2-1: Applicable Documents

3. XML READ/WRITE API SW OVERVIEW

3.1 OVERVIEW

The XML Read/Write API is a library to access SMOS products in Earth Explorer format. It will be mainly used by the developers of the SMOS components but it is designed and presented as a general public library. Moreover the XML Read/Write API is designed to fulfil the multi platform requirements even though some constraints may apply.

The XML Read/Write API uses the following libraries:

- Apache Xerces-C++ Parser: is a validating XML parser written in a portable subset of C++. Xerces-C++ makes it easy to give your application the ability to read and write XML data. A shared library is provided for parsing, generating, manipulating, and validating XML documents. See 3.6.3.3 for recommended version.
- Info-Zip, Info-UnZip libraries: are designed to be a free, general-purpose, legally unencumbered – that is, not covered by any patents – lossless data-compression libraries for use on virtually any computer hardware and operating system.

The interface provides users with the ability to use the bulk of the XML Read/Write API services inside their own applications. This interface is developed in C and interacts directly with all the libraries. This interface is supported in six other interfaces: CORE, XML, Binary, Ibox, FAD and Util.

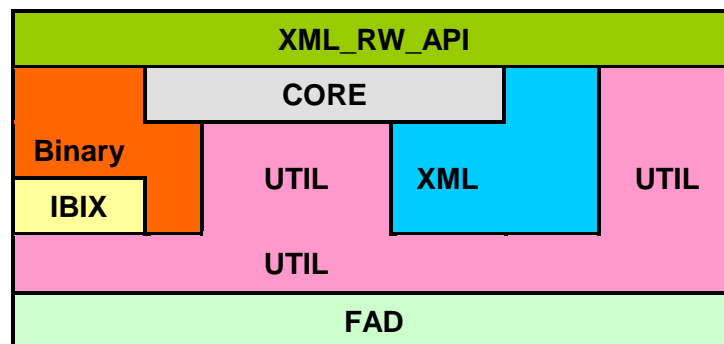


Figure 3-1: XML Read/Write API Library Interfaces

✓ **XML Read/Write API Interface:**

All the public methods and the functionalities of the application have to be published through a C codified interface. This library uses the services of the interfaces core, xml, binary and util. This is the API between the functionalities of the other interfaces, in C++ code, and the user.

✓ **CORE Interface:**

All the public methods and the functionalities of the application have to be published through a C++ codified interface. The interface is composed of methods for dealing with Earth Explorer File Format products of Level 0, 1, 2 and auxiliary data. These methods allow validating, reading, writing and creating Earth Explorer File Format products and auxiliary data. This library uses the services of the interfaces xml, binary and util.

✓ **XML Interface:**

The XML library is based on three interfaces for querying, generating and creating XML data files. This library uses the services of util interface. This interface is partially implemented using DOM parser from Xerces-C++ libraries.

✓ **Binary Interface:**

The XML Binary interface helps reading, writing and modifying binary information from the science products. This library uses the services of ibix and util interfaces.

✓ **Ibix Interface:**

This C++ code interface implements the reading and writing of binary data associated to BinX and XML schemas.

✓ **Files and directory (FAD) Interface:**

This C++ code interface implements the access to all the services for naming, renaming, copying, moving, deleting files through a local directory. It also grants access to all the services for packaging and un-packaging a product. This library does not include the services for product storage.

✓ **Util Interface:**

This C++ code interface implements the handler of schemas, exception, configuration files and other general services.

The XML Read/Write API Library is generated from the seven interfaces named above: XML READ/WRITE API, CORE, XML, Binary, Ibix, FAD and Util.

The XML Read/Write API is the interface between users and the other interfaces, these interfaces can not be installed or distributed apart from XML READ/WRITE API.

To access to the functionalities of the Library for external applications codified in any language different from C++ an interface in C has to be implemented.

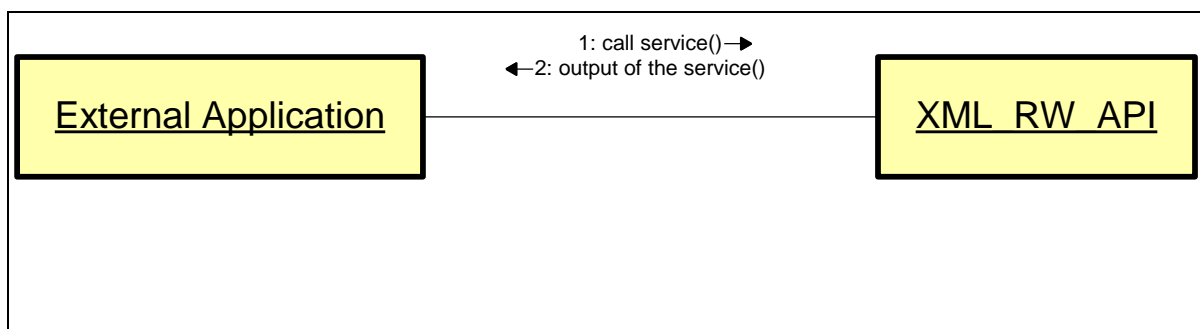


Figure 3-2: API interface collaboration diagram

The most important functionalities that the XML_RW_API offers are (see

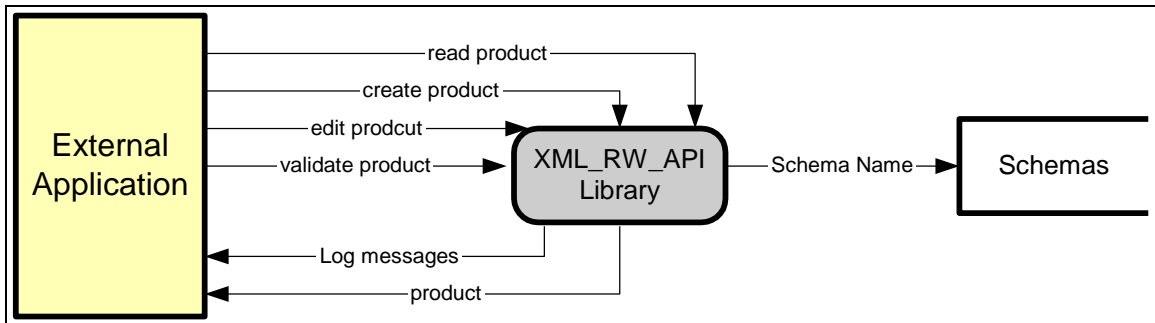


Figure 3-3):

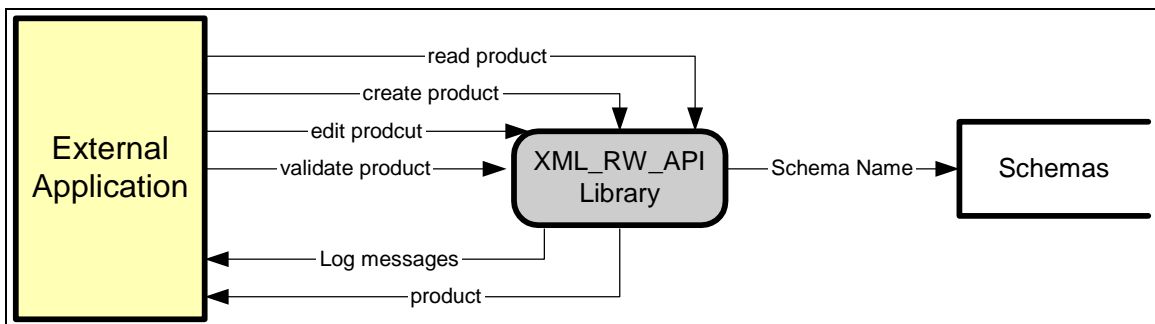


Figure 3-3: API interface context diagram

Whenever an application in Java has to access to the functionalities of the XML Read/Write API it will have to make use of JNI. (Figure 3-4)

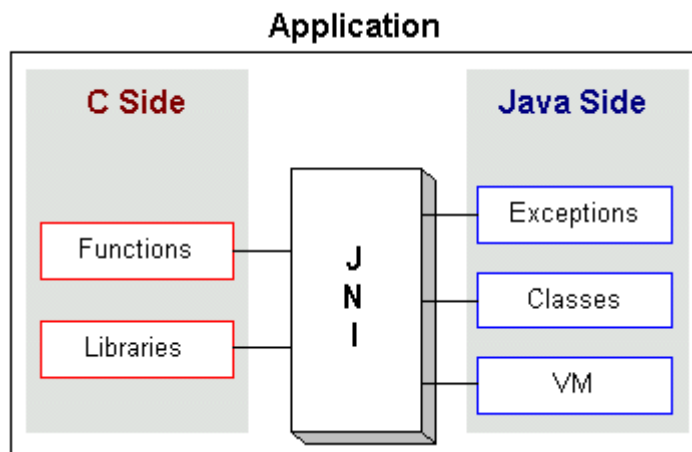


Figure 3-4: Java interface

Indra will not develop the Java I/F. It is the responsibility of the user that needs it.

3.2 SYSTEM REQUIREMENTS

The XML Read/Write API runs in the following environments:

- PC 32/64 bits under Linux
- PC 32 bits under Windows XP
- PPC 32 bits under MacOS X
- PC 32 bits under Cygwin
- PC 32 bits under Solaris

3.2.1 Linux

3.2.1.1 Hardware Requirements

The following bullets show the typical HW configuration required to use the XML Read/Write API under Linux:

- PC 32/64 bits with 2.60 GHz or higher processor clock speed recommended; 233 MHz minimum required (single or dual processor system); Intel Pentium/Celeron family, or AMD K6/Athlon/Duron family, or compatible processor recommended.
- 500 megabytes (MB) of RAM or higher recommended (128 MB minimum supported; may limit performance and some features)
- 1.5 gigabytes (GB) of available hard disk space (to store products).
- Network appropriate adapter for the type is recommended to update Xml and Binary Schemas.

3.2.1.2 Software Requirements

The source code has been compiled on an Intel PC 32 bits 2.60 GHz under Red Hat Enterprise Linux Server 5 and using the GNU *gcc* compiler. In summary, the software requirements are:

- Red Hat Enterprise Linux Server 5.
- GNU *gcc/g++* compiler version 4.1.1.
- Xerces-C++ for linux version 2.7.0.

The source code has been compiled on an Intel PC 64 bits under Linux Centos 7 and using the GNU *gcc* compiler. In summary, the software requirements are:

- Centos 7
- GNU *gcc/g++* compiler version 4.8.5.
- Xerces-C++ for linux 2.7.0

3.2.2 Windows XP

3.2.2.1 Hardware Requirements

The following bullets show the typical HW configuration required to use the XML Read/Write API under Windows XP:

- PC 32 bits with 2.60 GHz or higher processor clock speed recommended; 233 MHz minimum required (single or dual processor system); Intel Pentium/Celeron family, or AMD K6/Athlon/Duron family, or compatible processor recommended.
- 500 megabytes (MB) of RAM or higher recommended (128 MB minimum supported; may limit performance and some features)
- 1.5 gigabytes (GB) of available hard disk space (to store products).
- Network appropriate adapter for the type is recommended to update Xml and Binary Schemas.

3.2.2.2 Software Requirements

The source code has been compiled on an Intel PC 32 bits 2.60 GHz under Microsoft Windows XP and using the *Microsoft Visual C++* compiler.

In summary, the software requirements are:

- Microsoft Windows XP Operating System, version 2002 with Service Pack 1.
- Microsoft Visual C++ 6.0 Compiler (for linking the software to a C application)
- Xerces-C++ for Windows

3.2.3 MacOS X

3.2.3.1 Hardware Requirements

The following bullets show the typical HW configuration required to use the XML Read/Write API under MacOS X:

- PowerPC 32 bits G4 with 500 MHz or higher processor clock speed recommended; 233 MHz minimum required (single or dual processor system);
- 500 megabytes (MB) of RAM or higher recommended (128 MB minimum supported; may limit performance and some features)
- 1.5 gigabytes (GB) of available hard disk space (to store products).
- Network appropriate adapter for the type is recommended to update Xml and Binary Schemas.

3.2.3.2 Software Requirements

The source code will be compiled on a PowerPC 32 bits G4 500 MHz under MacOS X and using the GNU gcc compiler.

In summary, the software requirements are:

- MacOS X version 10.3.9 or later
- GNU gcc/g++ compiler version 3.3 or later versions
- Xerces-C++ for MacOS X

3.2.4 Cygwin

3.2.4.1 Hardware Requirements

The following bullets show the typical HW configuration required to use the XML Read/Write API under Windows XP:

- PC 32 bits with 2.60 GHz or higher processor clock speed recommended; 233 MHz minimum required (single or dual processor system); Intel Pentium/Celeron family, or AMD K6/Athlon/Duron family, or compatible processor recommended.
- 500 megabytes (MB) of RAM or higher recommended (128 MB minimum supported; may limit performance and some features)
- 1.5 gigabytes (GB) of available hard disk space (to store products).
- Network appropriate adapter for the type is recommended to update Xml and Binary Schemas.

3.2.4.2 Software Requirements

The source code has been compiled on an Intel PC 32 bits 2.60 GHz under Microsoft Windows XP and using the GNU gcc compiler.

In summary, the software requirements are:

- Microsoft Windows XP Operating System, version 2002 with Service Pack 1.
- Cygwin NT 5.1
- GNU gcc/g++ compiler version 3.4.4
- Xerces-C++ for Cygwin

3.2.5 Solaris

3.2.5.1 Hardware Requirements

The following bullets show the typical HW configuration required to use the XML Read/Write API under Sun Solaris:

- PC 32 bits with 2.60 GHz or higher processor clock speed recommended; 233 MHz minimum required (single or dual processor system); Intel Pentium/Celeron family, or AMD K6/Athlon/Duron family, or compatible processor recommended.
- 500 megabytes (MB) of RAM or higher recommended (128 MB minimum supported; may limit performance and some features)
- 1.5 gigabytes (GB) of available hard disk space (to store products).
- Network appropriate adapter for the type is recommended to update Xml and Binary Schemas.

3.2.5.2 Software Requirements

The source code has been compiled on an Intel PC 32 bits 2.60 GHz under Solaris 2.7 and using the GNU *gcc* compiler.

The software requirements are:

- Solaris 2.7 (or later) Operating System
- GNU *gcc/g++* compiler version 3.3.2
- Xerces-C++ for Unix

4. CONTENTS OF THE DELIVERY

The binary package naming should be `xml_rw_api_platform_2017-11-22_v05-00-00.zip`, where:

- platform could be:
 - o linux32.
 - o linux64.

The package must be downloaded and unzipped.

The library is given as a compressed file containing the following components:

SMOS XML Read / Write API.

- Projects
 - o bin
 - o lib
 - o xml_rw_api

SMOS XML auxiliary files:

- smos
 - o config
 - o products *
 - o schemas**
 - o tmp

* The “products” dir is empty. It is filled with products generated when “createTest” is executed.

** The “schemas” dir has one entry per product. This folder is empty. The user has to download the schemas and unzip it in this directory.

SMOS XML Read / Write API Test Tools.

- samples
 - o bin
 - o createTest
 - sources
 - install*
 - o editTest
 - sources
 - install*
 - o install*
 - o printAllTest
 - sources
 - install*
 - o readAllTest
 - sources
 - install*



- scripts
 - for_each_platform
 - createTest
 - createTest_RealProduct
 - editTest
 - readAllTest
 - validateTest
- validateTest
 - sources
 - install

The “scripts” dir has one entry per platform:

win32 (Windows XP)

unix (Linux, MacOS X, Solaris and Cygwin)

*Each “install” dir has one entry per platform:

win32 (Windows XP)

unix (Linux, MacOS X, Solaris and Cygwin)



indra



Ref. : SO-RN-IDR-GS-0256
Iss./Rev. : 1/0
Date: 22-Nov-2017

5. APPLICABLE XML SCHEMAS LIST

The applicable schemas to this XML R/W API version 05-00-00 are schemas_2007-11-19_v03-02-00 and later versions.

6. CHANGE LIST

Version	Date	Change Description
01-00-00	08-May-2006	First Delivery
01-01-00	20-Jun-2006	Updated Version
02-00-00	21-Jul-2006	Version for XML R/W API Factory Acceptance
02-01-00	11-Sep-2006	This version is released to correct most of the SPRs raised during XML R/W API Factory Acceptance.
02-02-00	17-Oct-2006	This version is released to correct some of the SPRs raised during XML R/W API Factory Acceptance.
02-02-01	27-Oct-2006	<ul style="list-style-type: none"> • Since this delivery, API will be distributed without schemas. • Updated to SO-TN-IDR-GS-0003 L0 Spec v1.8 2006-10-27.doc • Reduced memory leaks (i.e read one SM_TEST_MIR_SCLF1C product, which a size up to 500 MB, will not run out of the system memory). • Bug in getDataset() fixed. • install.sh script support static shared lib generation. • Added setMemoryByteOrder() and getMemoryByteOrder() methods. <ul style="list-style-type: none"> ○ setMemoryByteOrder() allows to set a fixed memory byte order. ○ getMemoryByteOrder() allows to get the current memory byte order. • TibixByte8 typedef to "signed char". • Added TibixString basic type. • Implemented copyProduct() method. • Removed ISO C++ "long long warning" on linux x86_64.
02-03-00	29-Nov-2006	<ul style="list-style-type: none"> • Reduced memory leaks (i.e read one SM_TEST_MIR_SCLF1C product, which a size up to 500 MB, will not run out of the system memory). • libxrwa shared library is in projects/bin • installTest.sh script support static shared exe generation. • setNodeData support struct types • Added getDataBlock_Schema() and getHeader_Schema() methods. • Added getErrorMessage() method
02-04-00	01-Dec-2006	Added macros:

Version	Date	Change Description
		<ul style="list-style-type: none"> • new_TxrwXXX(myType) • TxrwXXX myType = INITIALIZE_TxrwXXX • new_TxrwXXXPointer(myPointer) • new_constTxrwXXXPointer(myPointer)
02-04-01	05-Dec-2006	<ul style="list-style-type: none"> • getElementNode() returns row/col element instead of col/row element • Fixed editTest.c bug
03-00-00	22-Dec-2006	<ul style="list-style-type: none"> • Added getSchemaName(), getHeader_Schema() and getDatablock_Schema() methods. • Added getPartSize() method. • SchemaVersion attribute in readProduct(), createProduct(), editProduct() and validateProduct() should be: SM_CCCC_TTTTTTTTTT, where. <ul style="list-style-type: none"> ○ CCCC is the File Class or XXXX. ○ TTTTTTTTTT is the File Type.
03-00-01	12-Jan-2007	<ul style="list-style-type: none"> • Fixed createDataBlock() bug. • Fixed getElementNode() bug in multidimensional arrays and xml nodes. • Fixed getElementNodeByCount() bug in xml nodes. • Fixed getNextNode(), getPrevNode(), getParentNode(), getFirstChildNode() bug in xml nodes. • Fixed memory leaks.
03-00-02	12-Feb-2007	<ul style="list-style-type: none"> • Fixed readCounterFromFile() bug which corrupted the data obtained with getNodeData() for the first data value in a dsr
03-01-00	16-Feb-2007	<ul style="list-style-type: none"> • Added new methods: <ul style="list-style-type: none"> ○ getNodeOffset(): returns node offset relative to dataset record. ○ getNodeType(): returns node type. • Optimized performance: <ul style="list-style-type: none"> ○ simplified attributes handler. ○ getNode() method is optimized by using fast standard ANSI C library functions. • For xml nodes: <ul style="list-style-type: none"> ○ Solve problems with precision of float nodes. ○ Solve problems in function setNodeData.
03-02-00	23-Feb-2007	<ul style="list-style-type: none"> • Added one delivery compiled with 17checks option. • getPartSize works with: header (for xml and binary product), datablock (for xml and binary product), dataset

Version	Date	Change Description
		and datasetRecord. <ul style="list-style-type: none"> Added new methods: <ul style="list-style-type: none"> getFirstChildPart getNextPart getPrevPart getParentPart getElementPartData All of them can handle parts (not only nodes) and can return parts.
03-03-00	02-Mar-2007	<ul style="list-style-type: none"> Added deletePartData method. Updated class smosXMLNode as a DOMNode, unnecessary free memory for smosXMLNodes. Updated class smosXMLData: now it is not extended as a smosXMLNode. This class has a member smosXMLNode for representing header and datablock nodes. Updated all uses of header and datablock parts in functions of xml_rw_api. Solve problem creating lists when the name of the list is repeated. Solve problem setting data values when the value of the float was -0. Memory leaks.
03-04-00	20-Apr-2007	<ul style="list-style-type: none"> New class smosXMLFileParser is used for representing files parsed. New class smosXMLListFileParser is used for representing the list of files parsed. It is used for keeping a list of parsers of schemas while they are in use. In smosXMLNode: added new function getNodeype(). Updated getNodeValueXPath. Solve problem when a floating value is set. In smosXMLParser: changes when treat nodes for creating nodes structure, using smosXMLListFileParser and smosXMLFileParser to reduce processing time. Solve bug in validation function when attributes of the root tag are not defined or are bad defined. In xrwQuery: completed functionality of getNodeType.
03-05-00	19-Jun-2007	<ul style="list-style-type: none"> Updated xrwCore with treatment of extension in argument product path. New allowed extensions are ".EEF", ".XML" and ".HDR". Solve problem with access permission to products.

Version	Date	Change Description
		<ul style="list-style-type: none"> • More detailed description on header files. • Added getPartOffset() method. • Added getChecksumLikeInteger32() method. • Added ERROR_CODE_WRONG_ELEMENT_POSITION and ERROR_CODE_MATH_OVERFLOW errors. • Changed ERROR_CODE_MEMORY_EXCEPTION with ERROR_CODE_XERCES_MEMORY_EXCEPTION. • Updated Error Numbers. • Changed many ERROR_LEVEL_FATAL with ERROR_LEVEL_ERROR.
03-06-00	16-Aug-2007	<ul style="list-style-type: none"> • Added _FILE_OFFSET_BITS = 64 to smosProductFactory.cpp, fadIO.h, smosUtil.hpp, smosUtil.cpp, smosSchemaHandler, smosXMLNode.cpp (all source files where "sys/stat.h" is included) for solving problems in 32-bits system when products larger than 2GB are used (like MIR_GMAT product). • Added three new tags to xml_rw_api.lib_conf.xml and xml_rw_api.usr_conf.xml: <ul style="list-style-type: none"> ○ PREFIX_SCHEMA_NAMESPACE and PREFIX_PRODUCT_NAMESPACE allow to use different prefixes namespace in schemas and products. ○ LOWER_XML_EXTENSION allow to use ".xml" extension. ○ PRODUCT_TYPE_XML products are allowed to use ".xml" extension instead of ".XML" extension. • It is allowed to verify 19checksum and validate AUX_OEFPRE and AUX_OEFRES products. This kind of products can not be edited or created with this library.
03-06-01	30-Aug-2007	<ul style="list-style-type: none"> • Header and DataBlock are loaded independently. • Added getProductVersion() method.
03-07-00	24-Sep-2007	<ul style="list-style-type: none"> • This delivery only works from schemas version 2007-09-24_v02-02-00 on. • Optimized performance on XML files. • Create DBL file on createDatablock() method, instead of createProduct() method.
03-07-01	27-Sep-2007	<ul style="list-style-type: none"> • Fixed bug creating list of xml nodes.
03-07-02	11-Oct-2007	<ul style="list-style-type: none"> • Fixed bug calculating checksum. • Added getChecksumLikeUnsignedLong64() method.

Version	Date	Change Description
		<ul style="list-style-type: none"> Modify verifyChecksum() method to call internally to getChecksumLikeUnsignedLong64().
03-07-03	09-Nov-2007	<ul style="list-style-type: none"> Fixed concurrency bug Fixed bug searching identifiers in schemas_config file.
03-07-04	22-Feb-2008	<ul style="list-style-type: none"> Added line-feed. Added validation for products with noNamespaceSchemaLocation. Added “deletePart” function for binary Dataset and Dataset records. Fixed memory leaks.
03-07-05	30-Apr-2008	<ul style="list-style-type: none"> Fixed memory leak in method getDatasetRecord(). Fixed bug checking XML files well formed.
03-07-06	06-May-2008	<ul style="list-style-type: none"> Replaced MAX_ERROR_MSG_SIZE constant name by MAX_ERROR_MESSAGE_SIZE constant name.
03-07-07	28-May-2008	<ul style="list-style-type: none"> Fixed bug validating XML files. Fixed bug checking EE format.
03-07-08	04-Jul-2008	<ul style="list-style-type: none"> Fixed bug in deleteError() method

XML RW API versions for DPGS-V2 including the management of the schema versioning as indicated in [AD.5.]. These deliveries only work with schemas version 03-02-00 and later.

04-00-00	19-Nov-2007	<ul style="list-style-type: none"> Added checkProductSchemaVersion method to check the product schema version Added format for “count” attributes. Fixed memory leaks. Added treatment of scientific notation types. New treatment for schemas_config.xml (deleted binx references tags and “NAME” tags are not used by XML_RW_API). Deleted from xml_rw_api usr_conf.xml file tags that can not be edited by users. Added treatment of XMLV versioning for schemas.
04-00-01	24-Jan-2008	<ul style="list-style-type: none"> Fixed bug in AUX_SGLINT products. Fixed memory overwrite bug. Fixed bug in getNodeData function getting long and unsigned long values for Linux 32-bits. Fixed bug in setNodeData filling string values when the restriction of the schema was length, minLength. Added treatment of the “enumeration” restriction for schemas. Added treatment of the XML header and the XML datablock for the “getCount” and

Version	Date	Change Description
		“getCountByDimension” functions. <ul style="list-style-type: none"> The “SCHEMA_DIR” tag in xml_rw_api.usr_conf.xml allows relative and absolute paths. Fixed bug in checkProductSchemaVersion.
04-00-02	28-Jan-2008	<ul style="list-style-type: none"> Fixed bug writing datasets with zero dataset records. Fixed bug with absolute path in “SCHEMA_DIR” tag.
04-01-00	27-Feb-2008	<ul style="list-style-type: none"> Fixed bug writing string values in the XML products. Added validation for products with noNamespaceSchemaLocation. Added “deletePart” function for binary Dataset and Dataset records. Fixed memory leaks.
04-01-01	29-Feb-2008	<ul style="list-style-type: none"> Fixed bug writing string values in the XML products.
04-01-02	30-Apr-2008	<ul style="list-style-type: none"> Added FTP service to deal with remote schemas. Fixed memory leak in method getDatasetRecord(). Replaced MAX_ERROR_MSG_SIZE constant name by MAX_ERROR_MESSAGE_SIZE constant name. Added new method “setUserConfigFilePath”: this service enables to set the path for the xml_rw_api.usr_conf.xml configuration file by the user. Fixed bug checking well formed XML files.
04-01-03	29-May-2008	<ul style="list-style-type: none"> Fixed bug validating XML files. Fixed bug checking EE format. Fixed bug in getParent() method.
04-01-04	01-Aug-2008	<ul style="list-style-type: none"> Always checks that a xml file is well formed. Fixed bug in deleteError() method. Replaced delete by smosUtil__deleteObject. Replaced delete[] by smosUtil__deleteObject. Replaced deletes by smosUtil__deleteObjectArray. Replaced free by smosUtil__free. Simplified windows makefiles. Deleted regex libs on non window platforms.
04-01-05	26-Sep-2008	<ul style="list-style-type: none"> Fixed bug writing datasets with zero dataset records if it's not the last dataset. Fixed memory leak parsing a xml file.
04-02-00	12-Feb-2009	<ul style="list-style-type: none"> Added libinfo_unzip and libinfo_zip shared libraries Fixed bug validating subscription products. Fixed bug validating EEF_EX.xml reports.



Version	Date	Change Description
04-02-01	03-Apr-2009	<ul style="list-style-type: none">Fixed bug setting the counter offset in a variable array.
04-02-02	02-Oct-2009	<ul style="list-style-type: none">This is a fake delivery, implying no software modification. It is delivered to allow the SMOS Operations Team to exercise the installation procedure.
04-03-00	01-May-2014	<ul style="list-style-type: none">Fixed xercesc bug related to Header Size.
05-00-00	09-Nov-2017	<ul style="list-style-type: none">Rebuilt version for Centos 7, used xerces-c 2-7-0.Fixed errors related to g++ for Centos 7.