

DFDL4S++ Version 1.5.1 - Release Notes

1. NEW RELEASE DESCRIPTION

The DFDL4S++ (Version 1.5.1) is a C++ library providing a native interface to the functionality made available by DFDL4S Java library.

DFDL4S++ is a generic binary data binding library based on Data Format Description Language (DFDL). See more details at:

<https://eop-cfi.esa.int/index.php/applications/dfd4s>

1.1 Software and Hardware Requirements

DFDL4S++ is available for the following computer platforms:

- Windows XP / Vista / 7 (64-bit)
 - Microsoft Visual Studio 14.0 Express 64-bit
- Mac OS X Intel 10.9 or above (64-bit)
 - Apple LLVM v8.1.0 (clang-802.0.42) 64-bit
- Linux (64-bit)
 - GNU C++ Compiler (version 4.8.X) 64-bit

The DFDL4S++ library requires:

- Version 1.8 or higher of the Java Development kit
- 50+ Mb of hard disk space
- 2 GB RAM

1.2 Installation

A DFDL4S distribution package consists of one single archive (one per each supported platform):

- `dfd4s-1.5.1-win64.tar.gz`
- `dfd4s-1.5.1-mac64.tar.gz`
- `dfd4s-1.5.1-linux64.tar.gz`

To install DFDL4S++, extract the distribution package into a selected installation directory. Please refer to Section 3.2 of the DFDL4S++ Developer Manual for further information about the installation steps.

1.3 Documentation

For information related to DFDL4S++ library usage, please refer to the documentation of DFDL4S C++ API, in particular:

- Developer Manual: `DFDL4S++_Developers_Manual_S2G-DME-TEC-SUM113-1D.pdf`

All library documentation can be downloaded from the following location:

<https://eop-cfi.esa.int/Repo/PUBLIC/DOCUMENTATION/APPLICATIONS/DFDL4S/>

2. NEW FUNCTIONALITIES

This new release improves the library in the following aspects, with respect to v1.5.1:

- DFDL4S-AN-028: Consolidation of DFDL4S++ public API
 - A set of consistent public classes and methods is now available
 - Classes and methods related to internal processing have been removed from the public API
 - Harmonization of the DFDL4S (Java) API with DFDL4S++ (C++) API, so that the same functionality is provided in both languages
- Re-introduction of methods `childAt()` and `childAvailableCount()` in class `Element`, allowing an easy navigation of the `Element`'s tree
- Re-introduction of methods `getError()` and `getChildErrors()` in class `Element`, allowing to query the error status of interpreted binary data

3. CLOSED SPRS

N/A

4. ADDITIONAL COMMENTS AND KNOWN PROBLEMS

4.1 Mission Support

The DFDL4S distribution package already includes example mission configuration files (JAR archive containing XML files and schemas).

The latest version of the mission configuration files for all supported missions are available in the "Mission Files" section of the S2G website

(<http://eop-cfi.esa.int/index.php/applications/s2g-data-viewer/mission-files>)

The schema files in the website are compatible with the latest versions of the S2G application and the DFDL4S library.

4.2 Known Problems

- **[DFDL4S-ANR-0029]** Default constructors are not available

`Element` and `Document` cannot be initialised using default constructors, e.g.:

```
Document document;  
Element element;
```

Currently, objects of these types can be only initialised by assignment from objects created by the library itself. The library instantiates all objects that can then be modified.

- **[DFDL4S-ANR-0030]** `childAt()` method does not work properly in certain conditions

When used inside a loop that appends new `Elements` (e.g. `Space Packet`) to a document,

childAt() returns an incorrect Element:

```
for(int id = 0; id < size; ++id)
{
    dfdllib.appendElements( &document, &schema, data );
    Element e1 = document.childAt( id );
}
```

The workaround (see Example.cpp, included in the distribution package) is to create the file with all Elements, close the file, re-open it and set values to each field in the Element:

```
std::vector<unsigned char> data( size*size, 0 );
dfdllib.appendElements( &document_1, &schema, data );
document_1.close();
Document document_2 =
    dfdllib.interpretDocument( schema_file, filename );
```

5. REPORTING PROBLEMS

For any problems or questions please send an e-mail to the DFDL4S helpdesk:
dfd4s@eopp.esa.int