WHITEPAPER

IGT SOLUTIONS

Unlatch z/TPF data using DFDL

Introduction

This document on DFDL, provides an easily approachable description of the Data Format Description Language (DFDL or pronounced like daffodil), and should be used alongside to generate the DFDL schema for data defined in binary & text format etc.

Lots of the data in the world resides in files, which is not XML. This data is a mix of binary & text formats and doesn't have machine-readable description that can be shared with other computer systems. If the data is in XML or JSON then there are number of parsers & schema's available (language for describing the structure of data) for parsing the XML/JSON data that helps to integrate with other computer systems. Problem occurs in integrating the legacy systems with other computer systems where data is in binary, bit or text format emitted by z/TPF or C program, perhaps it is non-XML industry standard such x'12', EDIFACT and there are no standard parsers or schema's are available to describe the format/structure of data to aid integration. There are some standards available but they are too prescriptive and to use those data must be in specific format, but what if you already have the data and it's not in the prescribed format?

What is DFDL?

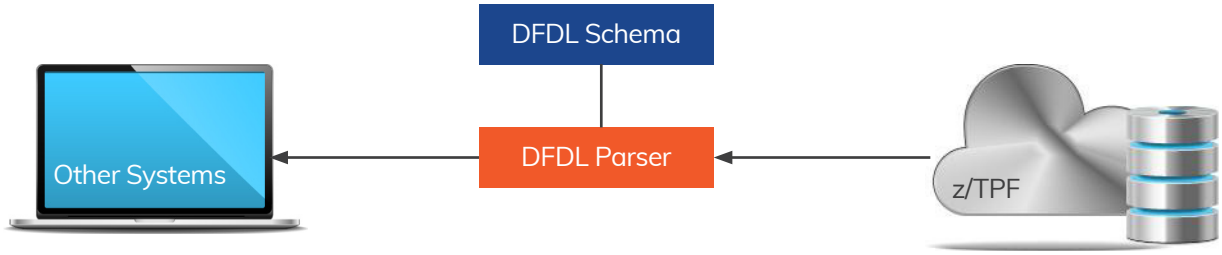
DFDL stands for Data Format Description Language and is powerful modelling language from Open Grid Forum, which describes the structure of text and binary data format, bit stream formats etc., all in a way that is independent of format itself. DFDL is not a data format and is based on XML schema with some DFDL annotations. It allows any binary or text data to be construe from its natal format and to be presented as an occurrence of information and vice versa.

When DFDL is modelled, the data can be parsed by DFDL processor (Such as IBM DFDL processor), for example, DFDL schema can be created for the binary data emitted by a legacy application, which will fully describe the binary data and then use this with IBM DFDL processer. The DFDL processor will use the DFDL schema to parse the data and creates the required data structure, so that data can be processed by other computer system or web services.

XML/JSON is to data is what DFDL is to assembler DSECTS & C structures.

DFDL doesn't dictate how data must be formatted instead it's the language for describing different kinds of data format. It's suitable for describing text, binary, Pipe delimited, scientific and numeric formats, legacy formats and bit stream formats etc. all in a way that is independent of DATA itself.

When data is modelled by DFDL then it can be parsed by DFDL parser to use with web services



Why DFDL

The need for data interchange was not evident when the programs were written initially in native formats. Today, DFDL is needed to standardize descriptions of data formats.

- There are plenty of parsers and schemas are available for XML/JSON but there is no universal standard for binary or text data. DFDL is designed to describe textual and binary data format while keeping focus on legacy systems files such as z/TPF files, z/TPFDF
- DFDL schemas help in converting z/TPFDF and other data into XML, JSON, and other consumable formats
- DFDL schemas of z/TPF records can be provided to the parsers to generate the structure for other systems to allow them to consume the z/TPF data
- Application intelligence can be embedded directly into the DFDL schema
- Number of tools came with their own proprietary method of describing data and these methods only works for the particular tool hence made the integration difficult across the systems

Use of DFDL in Today's z/TPF Environment

DFDL helps in describing the z/TPF binary and textual data enabling open technologies to access the z/TPF records. Written below is the list of top uses of DFDL in today's z/TPF environment

- DFDL can be used for describing the event message formats
- Using DFDL data can be converted to XML/JSON and other consumable formats
- Access z/TPF data by logical formats without knowing C Structure or Assembler DSECTS
- DFDL can be used for converting the XML/JSON back to native formats. This function is not supported yet on z/TPF
- DFDL can be used to provide the descriptive and meaningful names to describe the formats

Creating a DFDL schema file (Steps provided by IBM)

We can create the DFDL schema for z/TPF flat file records as well as z/TPFDF records.

1. ***z/TPF record/flat file schema –***

   Written below are the steps provided by IBM to generate the DFDL schema for z/TPF record (flat file)

   - Create a C structure & header file, or identify existing header file that defines the event data
   - The maketpf command along with specified DFDL target on the identified C structure and header will be issued to generate the DFDL schema
   - Extension for generated DFDL schema file will always be .dfdl.xsd
   - Move the generated file into your project for z/TPF descriptor definitions

2 ***z/TPFDF file schemas***

   Written below are the steps provided by IBM to generate the DFDL schema for z/TPFDF record

   - Toolkit wizard in IBM TPF toolkit to generate the DFDL for z/TPFDF record is not provided
   - ZUDFM DESCRIPTOR entry will be used on z/TPF to generate a DFDL Schema for z/TPFDF record
   - MLS information will be required to create DFDL schema using the above entry

3. ***Verify DFDL generated schema:***

   a) Data Types

   - DFDL takes the names from z/TPF (or z/TPFDF) DSECT and try to give best guess mappings of what this data type would mean in XML schema
   - Character data should be strings while non-numeric, non-character data should be hex binary

- Numeric data can be signed or unsigned but assembler DSECT doesn't provide any such information
- A byte consisting of various test bits could be changed to a bit-wise representation, 8 Boolean bit fields with a true/false assignment

b) Expressions

- DFDL discriminators for defining conditional data. Conditional data is the data that may not always be present or data based on the prior information such as union in C or data handled as ORG in assembler DSECT
- In DFDL formats corresponding to Union/ ORG data in z/TPF are handled as "Choice" or "Choice Branch". 'Choice' corresponds to variant record, multi-format record concept, where there is point of uncertainty & DFDL defines discriminators for the choice data and create the logic, which will be used to resolve the point of uncertainty
- String or Variable length field or arrays, where the length of the field/size of array is not defined. Here, DFDL expression will be used to make DFDL parser understand & calculate the length/size

Example: *Sample DFDL schema file*

References:
https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.13/gtpb1/tcevtdtdfdlschfl.html https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.13/gtpb1/tcevtdtdfdlschfl.html https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtpd1/tctadbidfdlconfig.html https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtps4/gtps4mst34.html https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtpd1/cdbdesxmp.html https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtpd1/tcdfdltpfrec.html

Using DFDL API's
http://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtps6/usdfdlapi.html

PUT 13 update
http://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtpm2/m2chput13.html

Supported DFDL v1.0 Specification
http://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtps6/cdfdlspec.html

IBM z/TPF Webinars
https://www.youtube.com/channel/UCkbauHDRSo6QV-2sB7UsXpg

Conclusion

DFDL is powerful tool for describing the structure of binary and textual data. It provides the ability to convert the z/TPF binary data to XML/JSON and will continue to be enhanced and leveraged by z/TPF.

Today, z/TPF application code is required to remote access the data residing on z/TPF from remote platform in open environment, which involves parsing of input/output request and response, database access for e.g.

z/TPFDF API's, which increases the time to market & cost of maintaining and enhancing these custom applications for new business.

With the advent of MongoDB and DFDL, binary & textual (z/TPFDF) data can be described using the DFDL and further utilising MongoDB z/TPF data can be accessed directly which will reduce the cost and time to market.

IGT Solutions (IGT) is a leading BPM, Technology and Digital Services and Solutions Company committed to deliver innovation and business excellence across the entire spectrum of Travel, Transportation and Hospitality domain.

Established in 1998, with 100% focused on the Travel industry, we have more than 70 marquee customers globally. IGT serves 4 in top 5 Airlines, 5 out of Top 5 Travel Companies, 4 out of Top 5 Hospitality companies. We provide digital contact center services, travel technology and innovative digital services and solutions for 100+ travel processes including Reservations and Sales, Customer Service, IROPS Management, Baggage Helpdesk, Crew Helpdesk, Chatbots, Robotic Process Automation, Travel Analytics and Social Media Services.

IGT SOLUTIONS

IGT Solutions Pvt. Ltd.

Echelon Building, Plot No. 49,
Sector-32, Gurgaon - 122 001,
Haryana, India

T +91 (0)124 458 7000
F +91 (0)124 458 7198
mktg@igtsolutions.com
www.igtsolutions.com