

z/TPF DFDL Serializer Support for CSV

2024 TPF Users Group Conference

May 5-8, New Orleans, LA

Web Services Subcommittee

—

Abigail DeSantis

DFDL adopted by ISO!

The DFDL specification has been adopted by ISO as **ISO/IEC 23415:2024**.

<https://www.iso.org/standard/87444.html>

The spec will still be maintained and available for free on OGF:

<https://ogf.org/documents/GFD.207.pdf>

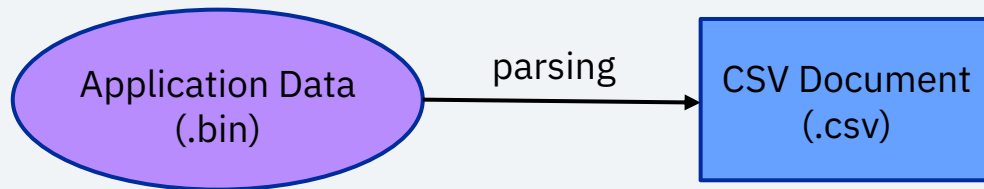
Being adopted by ISO advances the DFDL standard to be more widely accepted among a global audience.

Problem Statement

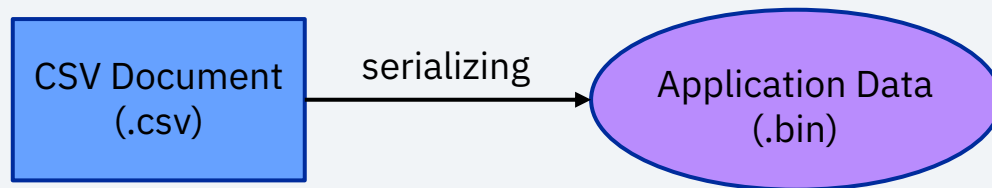
To convert a CSV document into application data, code must be written for each CSV document with unique columns to parse it into a format usable by the application. If the columns change, the code will need to be changed.

As-Is Background

There is functionality that uses DFDL to take application data (.bin) and write it into a CSV document (.csv), but no functionality to consume a CSV document and convert it to application data.

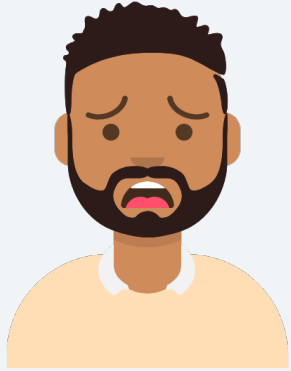


A CSV document *can* be parsed from binary data by using DFDL



A CSV document *cannot* be serialized into binary data by using DFDL.

As-Is User Story



Zach

Application developer

My application needs to consume a new CSV file. Now I need to write a bunch of code to be able to parse the data.

I will also need to transfer a set of CSV files to my test system to test various conditions to ensure my code behaves properly.

This is going to be a lot of work!

Pain Points

- A parser must be created for each unique CSV data to convert it to a format usable by the application.
- Any time columns change, the parser code must be updated.
- A separate parser is needed for each version of CSV data.

Value Statement

An application programmer can use DFDL to greatly reduce the amount of code and testing time needed to consume CSV documents on z/TPF.

To-Be User Story

- All that is needed is to create a DFDL document that describes the CSV document. The DFDL serializer API, `tpf_dfdl_serializeDoc`, can be used to consume the CSV document and create a binary format usable by the application.
- No additional code required!
- If the CSV document is changed, only the DFDL needs to be updated. The serializer is able to handle the changes.

To-Be User Story



Zach
Application developer

My application needs to consume a new CSV file. Now all I need to do is create a DFDL document that describes the CSV document and call the DFDL serializer API. So simple!

And to test my DFDL file, I can use the z/TPF offline utility (`tpfdatamap`) on Linux on IBM Z, which will save me a lot of time.

Technical Details – PJ47158 (April 2024)

Overview

To serialize a CSV document, call the [tpf_dfdl_serializeDoc](#) function with `DFDLFormat` set to `TPF_DFDL_CSV`.

The following options are supported only for CSV documents:

- `TPF_DFDL_ROWVAL` – Each element and its corresponding value appear on the same row.
- `TPF_DFDL_STRICT` – If the document contains data not defined in the DFDL, processing stops, and an error is returned.
- `TPF_DFDL_XTAGS` – All element tags are excluded from the document.

Technical Details – PJ47158 (April 2024)

CSV documents with no options specified

flight_number,origin,destination,gate,departure_time

WB1570,JFK,MSY,30A,"2024-05-04 09:53:00"

firstname,lastname,group,seat,carry_on,checked_bags

Roy,Cooper,2,36B,true,0

Jane,McGavin,3,12A,false,2

(complex element headers)

(complex element data)

(complex array headers)

(array data element 1)

(array data element 2)

| | A | B | C | D | E | F |
|---|---------------|----------|-------------|------|----------------|--------------|
| 1 | flight_number | origin | destination | gate | departure_time | |
| 2 | WB1570 | JFK | MSY | 30A | 5/4/2024 9:53 | |
| 3 | firstname | lastname | group | seat | carry_on | checked_bags |
| 4 | Roy | Cooper | 2 | 36B | TRUE | 0 |
| 5 | Jane | McGavin | 3 | 12A | FALSE | 2 |

Technical Details – PJ47158 (April 2024)

CSV documents with TPF_DFDL_ROWVAL option specified

flight_number,WB1570

origin,JFK

destination,MSY

gate,30A

departure_time,"2024-05-04 09:53:00"

firstname,Roy

lastname,Cooper

group,2

seat,36B

carry_on,true

checked_bags,0

firstname,Jane

lastname,McGavin

group,3

seat,12A

carry_on,false

checked_bags,2

(headers)

(complex element data)

(complex array element 1 data)

(complex array element 2 data)

| | A | B |
|----|----------------|---------------|
| 1 | flight_number | WB1570 |
| 2 | origin | JFK |
| 3 | destination | MSY |
| 4 | gate | 30A |
| 5 | departure_time | 5/4/2024 9:53 |
| 6 | firstname | Roy |
| 7 | lastname | Cooper |
| 8 | group | 2 |
| 9 | seat | 36B |
| 10 | carry_on | TRUE |
| 11 | checked_bags | 0 |
| 12 | firstname | Jane |
| 13 | lastname | McGavin |
| 14 | group | 3 |
| 15 | seat | 12A |
| 16 | carry_on | FALSE |
| 17 | checked_bags | 2 |

Technical Details – PJ47158 (April 2024)

CSV documents with TPF_DFDL_XTAGS option specified

WB1570,JFK,MSY,30A,"2024-05-04 09:53:00" (complex element data)
Roy,Cooper,2,36B,true,0 (array data element 1)
Jane,McGavin,3,12A,false,2 (array data element 2)

| | A | B | C | D | E | F |
|---|--------|---------|-----|-------|---------------|---|
| 1 | WB1570 | JFK | MSY | 30A | 5/4/2024 9:53 | |
| 2 | Roy | Cooper | | 2 36B | TRUE | 0 |
| 3 | Jane | McGavin | | 3 12A | FALSE | 2 |

Technical Details – PJ47158 (April 2024)

Additional Considerations

- Double quotation marks are required for any string containing a comma (,) or a newline character (\n). Double quotes are optional for all other cases. In all cases, double quotes will be stripped before storing the string value. If no double quotes are used, the string will be stored as provided.
- CSV documents can be tested with the DFDL serialize offline utility.
- The CSV document must be UTF-8.

Technical Details – PJ47158 (April 2024)

Version Control

- All new fields added must be non-required.
- If the CSV document is a previous version and does not contain the new fields, DFDL will create a single data format with default values for the fields not in the document.
- A default value must be set for the new field.

Conclusion

PJ47158 (April 2024): DFDL serializer support for CSV

- The DFDL serializer for CSV reduces the amount of code needed to convert each unique CSV data into a format usable by the application.
- Before this enhancement, code would need to be written specific to each unique CSV data to convert CSV data into binary format. If the CSV document was altered, the code for parsing would need to be updated.
- With this enhancement, we just use the DFDL serializer API with the CSV document and DFDL and the serializer will create the binary content.
- Reduces testing time with the use of *tpfdatamap* on Linux on IBM Z.
- Reduces maintenance costs by managing unique CSV document versions with only updates to the DFDL document.

Thank you

© Copyright IBM Corporation 2022. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

